21世纪高等学校规划教材 | 计算机科学与技术





计算机控制技术与系统仿真

翟天嵩 主编
刘忠超 米建伟 副主编

清华大学出版社

21世纪高等学校规划教材·计算机科学与技术

计算机控制技术与系统仿真

翟天嵩　刘忠超　米建伟　等编著

清华大学出版社
北　京

内 容 简 介

本书从计算机控制系统的设计与实现的基础知识出发，系统阐述了计算机控制技术中硬件、软件、控制算法的设计与实现方法。不仅对计算机控制系统的设计、分析、校正和应用的基础内容进行了总结和讲解，而且介绍了利用 Proteus 软件进行硬件电路的仿真，利用 MATLAB 软件进行控制算法的仿真与分析，以及使用组态王软件进行控制系统人机界面的设计。在介绍设计方法的同时融入典型实例分析，力求读者在学习专业知识的过程中，掌握一些辅助性的软件工具，提高学习兴趣和学习效率，也为教师提供一种新的教学思路。

本书可作为高等院校自动化、电气工程及其自动化、计算机应用、测控技术、机电一体化等专业应用型本科生的授课教材，也可作为相关专业的研究生教材，同时可作为控制类专业教师的辅助教材，以及从事计算机控制技术的工程人员的工具书。

图书在版编目（CIP）数据

计算机控制技术与系统仿真/翟天嵩主编.--北京：清华大学出版社，2012.9（2016.1 重印）
（21 世纪高等学校规划教材・计算机科学与技术）
ISBN 978-7-302-29383-5

Ⅰ．①计…　Ⅱ．①翟…　Ⅲ．①计算机控制－高等学校－教材 ②系统仿真－高等学校－教材　Ⅳ．①TP273 ②TP391.9

中国版本图书馆 CIP 数据核字（2012）第 158593 号

责任编辑：高买花　薛　阳
封面设计：傅瑞学
责任校对：焦丽丽
责任印制：何　芊

出版发行：清华大学出版社
　网　　址：http://www.tup.com.cn，http://www.wqbook.com
　地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
　社 总 机：010-62770175　　**邮　　购**：010-62786544
　投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
　质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
　课件下载：http://www.tup.com.cn，010-62795954
印 装 者：虎彩印艺股份有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　**印　张**：21　　**字　　数**：511 千字
版　　次：2012 年 9 月第 1 版　　**印　　次**：2016 年 1 月第2 次印刷
印　　数：3001～3500
定　　价：34.50 元

产品编号：042373-01

编审委员会成员

（按地区排序）

浙江大学	吴朝晖	教授
	李善平	教授
扬州大学	李　云	教授
南京大学	骆　斌	教授
	黄　强	副教授
南京航空航天大学	黄志球	教授
	秦小麟	教授
南京理工大学	张功萱	教授
南京邮电学院	朱秀昌	教授
苏州大学	王宜怀	教授
	陈建明	副教授
江苏大学	鲍可进	教授
中国矿业大学	张　艳	教授
武汉大学	何炎祥	教授
华中科技大学	刘乐善	教授
中南财经政法大学	刘腾红	教授
华中师范大学	叶俊民	教授
	郑世珏	教授
	陈　利	教授
江汉大学	颜　彬	教授
国防科技大学	赵克佳	教授
	邹北骥	教授
中南大学	刘卫国	教授
湖南大学	林亚平	教授
西安交通大学	沈钧毅	教授
	齐　勇	教授
长安大学	巨永锋	教授
哈尔滨工业大学	郭茂祖	教授
吉林大学	徐一平	教授
	毕　强	教授
山东大学	孟祥旭	教授
	郝兴伟	教授
厦门大学	冯少荣	教授
厦门大学嘉庚学院	张思民	教授
云南大学	刘惟一	教授
电子科技大学	刘乃琦	教授
	罗　蕾	教授
成都理工大学	蔡　淮	教授
	于　春	副教授
西南交通大学	曾华燊	教授

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程”（简称“质量工程”），通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展，顺应并符合 21 世纪教学发展的规律，代表并引领课程发展的趋势和方向）、示范性（教材所体现的课程体系具有较广泛的辐射性和示范性）和一定的前瞻性。教材由个人申报或各校推荐（通过所在高校的“编委会”成员推荐），经“编委会”认真评审，最后由清华大学出版

社审定出版。

目前，针对计算机类和电子信息类相关专业成立了两个“编委会”，即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括：

（1）21世纪高等学校规划教材·计算机应用——高等学校各类专业，特别是非计算机专业的计算机应用类教材。

（2）21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

（3）21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

（4）21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

（5）21世纪高等学校规划教材·信息管理与信息系统。

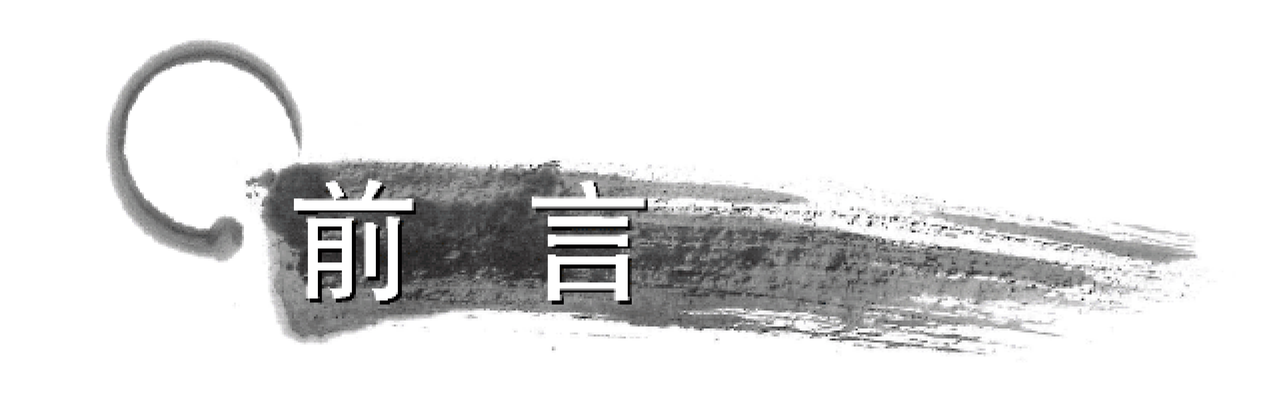
（6）21世纪高等学校规划教材·财经管理与应用。

（7）21世纪高等学校规划教材·电子商务。

（8）21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
联系人：魏江江
E-mail：weijj@tup.tsinghua.edu.cn



前言

计算机控制技术是高等院校自动化、电气工程及其自动化、计算机应用、测控技术、机电一体化等专业的专业主干课程，也是实现工业生产自动化的基础技术，是计算机技术、自动化理论与工业生产控制紧密结合的产物。涉及到电子技术、控制理论、传感器技术、通信技术和系统工程等诸多领域。其应用越来越广泛，并逐渐向智能化、网络化和集成化的方向发展。

本书注重基础性、系统性和实用性，全面、系统地总结了计算机控制系统的理论，以及系统硬件、软件和控制策略的设计与实现方法，从培养应用型本科人才出发，阐述基本理论的同时兼顾工程实践应用；讨论了怎样将功能强大的 EDA 工具软件——Proteus——用于硬件电路的设计与仿真；还将 MATLAB 和 Simulink 用于计算机控制系统的分析和控制器的设计；通过大量实例和仿真方法的研究，加强学生的自学能力、创新能力和实践能力的培养与锻炼，引导学生能够快速地进行计算机控制系统的分析、设计和实现。另外，掌握恰当的仿真方式还可以提高教师教学的生动性，提高学生学习的兴趣和效率。本书最后还给出了部分实验内容，便于选择该教材的教师在教学时参考使用。

本书共分为 11 章，第 1 章主要讲述计算机控制系统的基本概念、系统的典型形式、计算机控制技术的发展情况，以及该课程学习的要点；第 2 章通过一个计算机控制系统的应用范例，使读者对计算机控制系统有一个整体的了解；第 3 章阐述计算机控制系统的硬件设计技术；第 4 章讨论硬件仿真技术，并着重介绍使用 Proteus 软件进行硬件电路的仿真；第 5 章主要讨论计算机控制系统的控制算法；第 6 章通过对 MATLAB 软件的介绍，讨论如何进行控制系统的仿真与分析；第 7 章主要研究数字程序控制技术，以步进电机为驱动设备，重点介绍插补算法、步进进给方式等；第 8 章讨论计算机控制系统软件设计方法、OPC 技术和数据处理技术；第 9 章主要介绍使用组态软件进行系统监控界面的设计；第 10 章讨论计算机控制系统的设计原则、方法，以及设计实例；第 11 章是课程实验，可供实验教学选用。

书中所有的实例都经过调试，有很强的实践参考价值。

本书编写过程中，南阳理工学院翟天嵩任主编，南阳理工学院刘忠超和西安电子科技大学米建伟任副主编。第 1、3、4 章由翟天嵩编写，第 2 章由米建伟编写，第 5、9 章由刘忠超编写，第 6 章和第 8 章的 8.1 节、8.2 节由张丹编写，第 7 章和第 11 章由熊雷编写，第 8 章的 8.3 节和 8.4 节由崔玉连编写，第 10 章由刘勇军编写。

本书为南阳理工学院“优秀教材”建设项目，由高有堂教授主审，在此表示感谢。

计算机控制技术是一项涵盖面宽、应用性强，且发展较快的技术，编写过程中，我们尽量做到知识严谨、技术实用，但鉴于编著者水平有限，书中难免有疏漏之处，敬请广大读者批评指正。

编　者

2012 年 4 月


目 录

第1章 绪论

计算机控制系统是利用计算机(包括其他多种控制器)来实现生产过程或运动对象自动控制的系统。因此,计算机控制技术是一门综合性的技术,是以电子技术、自动控制技术、计算机应用技术为基础,综合接口技术、可编程控制技术、单片机技术、计算机网络技术、通信技术、自动化检测技术、显示技术、软件工程技术等,最终实现生产技术的精密化、生产设备的信息化、生产过程的自动化,以及机电控制系统的最佳化的一项专业学科。

1.1 计算机控制系统的基本概念

计算机控制系统的主要功能可归纳为以下三个方面。

(1) 实时数据采集:利用检测、变送装置,实时地对被控制量的瞬时值进行采集和输入。

(2) 实时控制决策:对采集、输入的数据进行比较、分析和处理,并按照预定的控制策略(算法)进行运算,产生决策信号。

(3) 实时控制输出:根据决策信号实时控制执行机构,实现对被控制对象的控制任务。

因此,一个基本的计算机控制系统主要包括:控制器(常用工业控制计算机,简称工控机)、接口电路、执行机构、被控制对象、检测机构、显示设备,以及相应的控制软件、监控程序等。

1.1.1 计算机控制系统的基本组成

计算机控制系统主要由控制系统和被控生产过程对象两大部分组成。控制系统又包括硬件和所匹配的软件。

1. 控制系统的硬件结构

如图1.1所示,控制系统的硬件基本结构包括:主控设备(控制计算机)、被控制对象(生产过程),以及输入和输出设备。

输入电路又包括模拟量输入通道和数字量输入通道。模拟量输入通道主要利用变送器测量被控制对象的相关模拟量数据(如温度、压力、流量等),经采样开关采样,然后进行模拟量到数字量的转换(A/D转换),变换成适于计算机处理的信号,经接口电路送入控制计算机进行数据处理。数字量输入通道主要用于输入一些开关量信号并且在控制对象和控制计算机之间需进行有效的隔离,如光电隔离等。

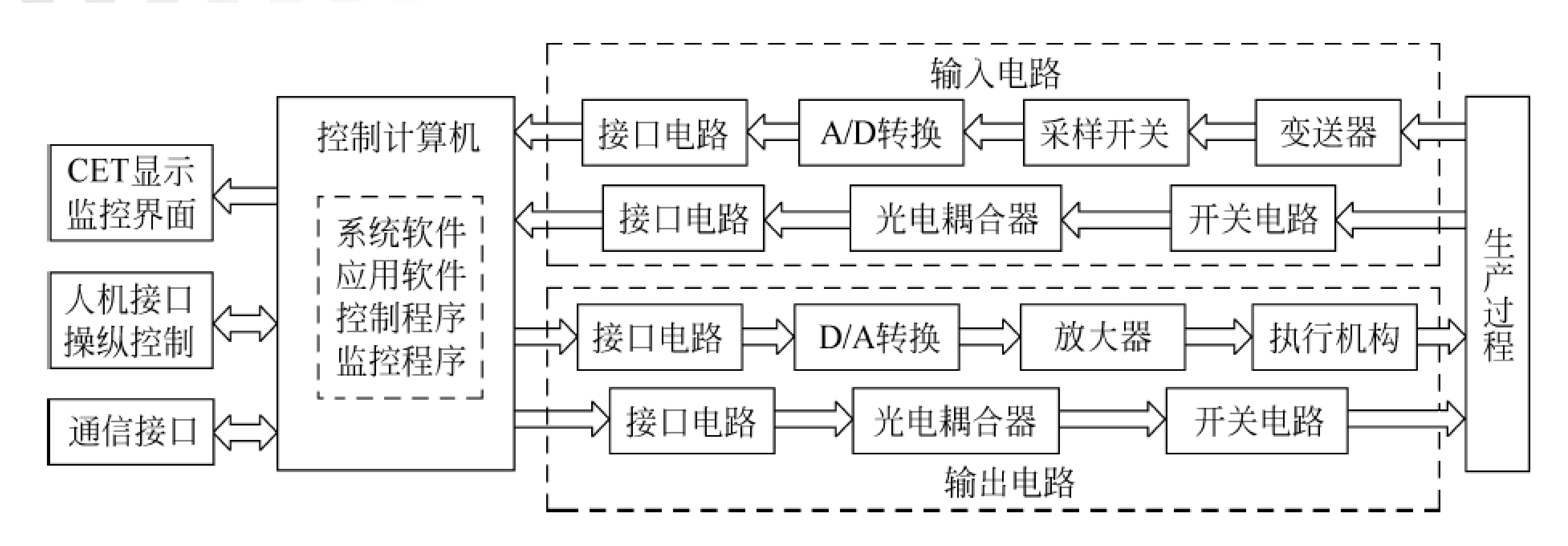


图 1.1 计算机控制系统基本结构框图

输出电路同样包括模拟量输出通道和数字量输出通道。模拟量输出通道主要用于控制模拟量输出的执行机构，因此从控制计算机输出经接口电路的数字信号，需先进行数字量到模拟量的转换（D/A 转换），然后进行放大处理（D/A 转换后的信号一般驱动能力达不到控制的要求），最后用于控制执行机构。而数字量输出通道主要用于输出一些开关量信号，在控制器和被控制对象之间也需要进行必要的隔离。

另外，为了及时了解被控制对象的运行情况，通过软件设计，控制计算机可以通过显示器提供一个监控界面，如图 1.2 所示。监控界面在显示器上可实时显示当前被控制变量的实时数据、实时曲线、历史曲线等，同时也可以通过监控界面上所设计的按钮、选择开关或数据输入项等进行控制信号的输入，对生产过程进行实时控制。如图 1.2 所示为一个电加热锅炉水温控制系统监控界面。界面上方实时显示当前的温度变化曲线和设定值曲线，可以用来观察曲线的变化趋势，以及与理论曲线间的关系；中间依次显示启动/停止按钮、当前温度的数值，以及急停键；最下方是显示用于控制的控制参数。有了这个界面，控制操作人员就可以方便地获取控制参数，也能更好、更直接地实现对系统的监控。

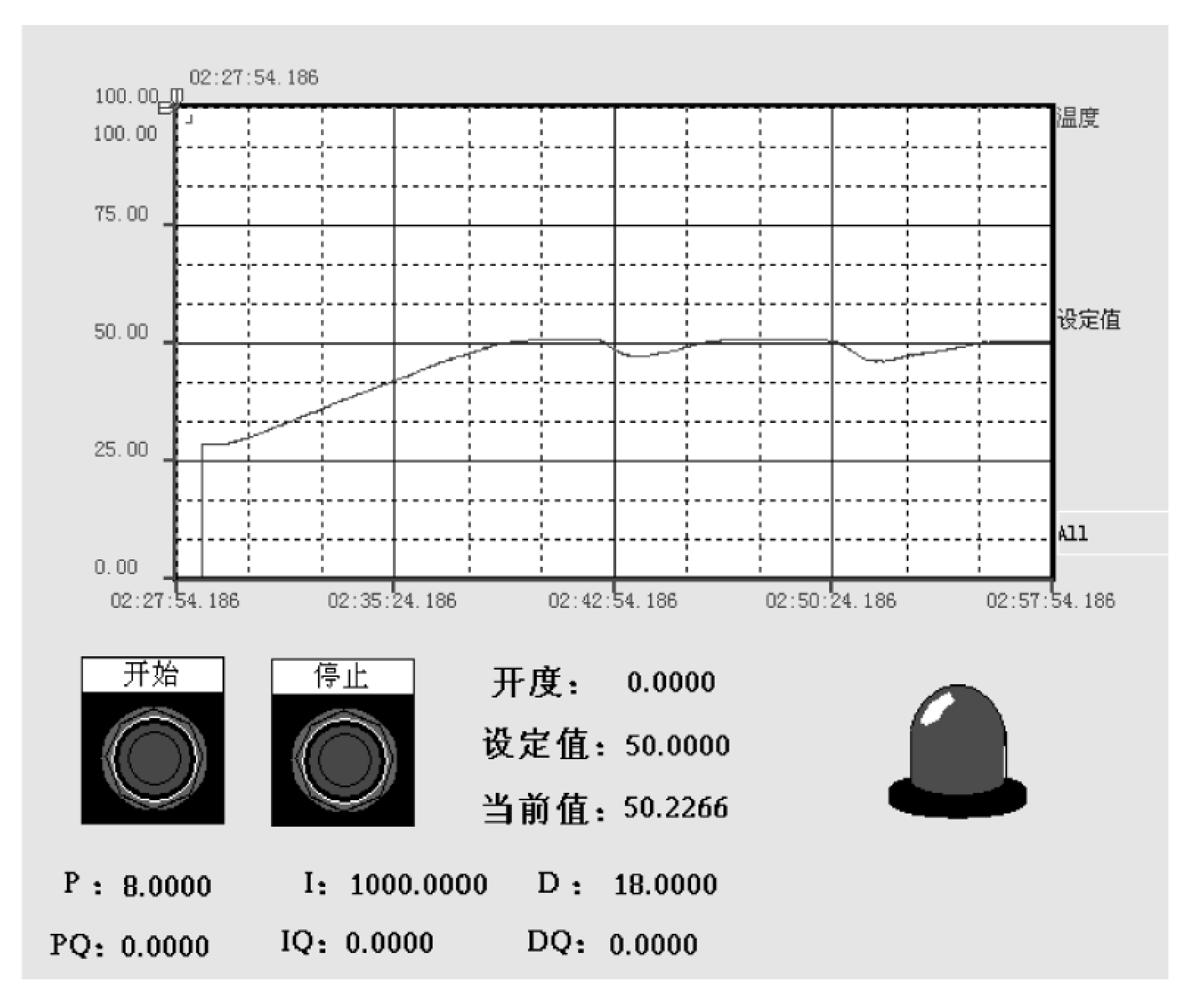


图 1.2 电加热锅炉水温控制系统监控界面

较为复杂的系统，还可以构成基于各类工业总线的工业控制网络，通过工业网络来实现较为复杂的系统控制。如图 1.3 所示为工业控制网络示意图。

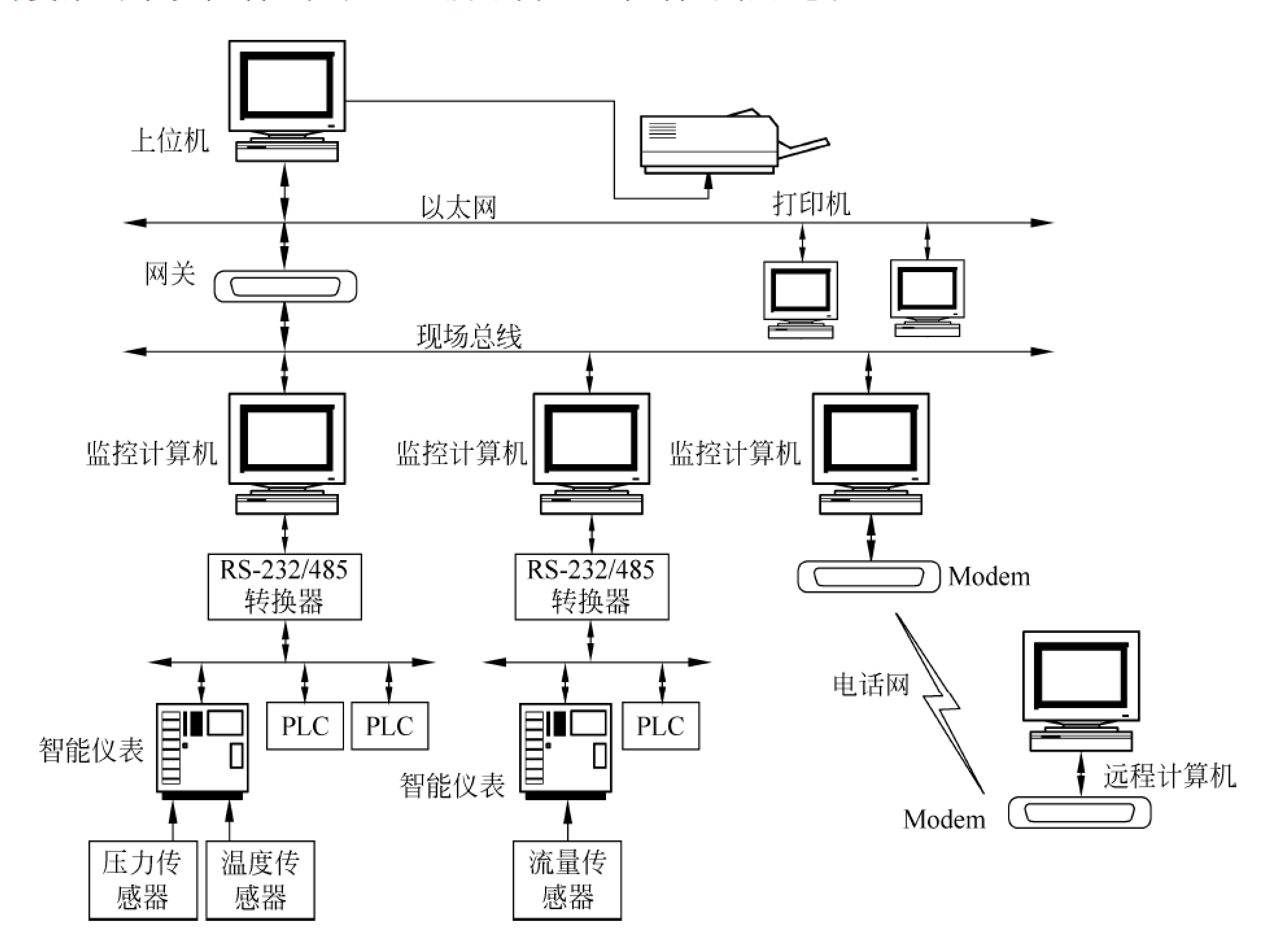


图 1.3　工业控制网络示意图

2. 控制系统的软件结构

控制系统的软件是计算机控制系统的灵魂。整个计算机系统的动作，都是在软件的指挥下协调进行的。软件系统主要包括系统软件、应用软件和数据库三部分。

系统软件是指控制和协调计算机及外部设备，支持应用软件开发和运行的系统，主要功能是调度、监控和维护计算机系统，使计算机使用者将计算机当作一个整体而不需要顾及到底层每个硬件是如何工作的。主要包括：操作系统、程序设计语言、语言处理程序(编译程序)、数据库管理程序、通信网络软件、诊断程序、系统辅助处理程序等。

应用软件是由用户根据需要解决的各种实际问题而编写的程序。计算机控制系统的应用软件主要有：监视程序(包括被控制量的巡回检测程序、数据处理程序、上下限检查及报警程序、操作面板服务程序、数字滤波及标度变换程序等)、控制算法程序、过程控制程序、人机接口程序、输入/输出程序等。

数据库及数据库管理系统主要用于数据的管理、存档和检索。数据库包括历史数据库和实时数据库。相应的软件设计指如何建立数据库以及如何查询、显示、调用和修改数据等。

1.1.2　计算机控制技术所涉及的主要问题

通过分析计算机控制系统的基本结构可以发现计算机控制技术所涉及的主要问题包括

以下几个方面。

1. 计算机应用技术

本课程所采用的主要控制器是工业控制计算机,简称工控机,这就需要能非常熟悉计算机的硬件结构;并能熟练掌握常用计算机软件操作,至少熟练掌握一门编程语言,具有一定分析问题和解决问题的能力;能够通过硬件设计、接口设计和软件编程进行系统设计。

2. 信号采集

信号采集即生产过程被控制参数的检测。要使得系统能够以一定的控制精度稳定运行,必须借助一定的传感器、变送器进行生产过程的参数检测,来反映系统当前的运行情况,从而与给定信号进行比较,形成偏差,以更好完成系统的控制。检测到的信号在传送之前,首先要转换成一种标准形式,通常,将传感器的输出信号转换成 4～20mA 的电流(用于较远距离的传输)或是 1～5V 的电压(用于近距离传输)。

开关状态的信息,如阀门是否关闭、运动装置是否到达位置、水位是否达到等,常通过一些触点的通断或 TTL 电平的变化来表示。

3. 控制与决策

计算机控制系统的控制器属于数字控制器,用于偏差的计算和偏差的处理。此种控制器的设计方法包括模拟化设计方法和数字化设计方法。常用的控制算法有:数字 PID 算法、改进的数字 PID 算法、最小拍控制算法、史密斯预估算法、达林算法、模糊算法,以及解耦算法等。可根据被控制变量的变化规律设计与之相匹配的控制策略。

4. 软件设计

系统软件的设计主要包括两个方面,一是控制软件的实现,用于实现信号采集、信号处理、偏差处理、控制算法、执行器驱动、手动/自动控制等;另一是监控软件的设计,提供友好的人机界面,不仅可以使操作控制人员较全面的了解系统运行的实时数据、历史数据、分析报表、运行的进程等,而且可以对系统的控制操作更方便、直观。

5. 通信技术

当控制系统采用分散式控制结构时,往往由多台计算机分级进行系统工作过程的控制,这就需要各级计算机之间能够及时、准确的交换信息。采用恰当的通信方式,可以使信息的传送更通畅。

6. 显示技术

计算机控制系统中常采用的显示形式多种多样,主要包括:报警指示、信号状态显示(Light Emitting Diode,LED)、数码管显示、电脑显示屏显示、液晶显示屏显示、触摸屏等。

7. 电机驱动技术

电机的种类很多,计算机控制系统中的电机驱动电路可根据所选电机的类型进行自行

设计，也可以购买电机生产厂家的配套产品。如步进电机可选择配套的步进驱动电源；直流伺服电机可选择直流伺服驱动器；交流电机可选用变频器控制；交流伺服电机可选配交流伺服驱动器等。

1.1.3　计算机控制技术的应用领域

随着科学技术的高速发展，计算机控制技术在工业生产、航空航天、交通管理、楼宇自动化、智能机器人，甚至日常生活等各个领域的应用都有了普遍的提升。

制药行业就融合了现代科技手段和高新技术中的精华，为传统的制药企业带来了一场新的革命。例如，空气的洁净度控制主要通过过滤器设备和室内达到一定的换气量来实现。过滤器的性能好坏及是否堵塞，直接影响到空气的洁净度。如果靠人为的观察去达到这一目标是不能满足要求的，还是要借助于科学仪器来检测，一般采用压差式仪表。可采用现场式和远程式。远程式仪表配合计算机网络系统即可实现实时监测和控制。又如温度的控制，以往是靠手动控制冷热源的流量来实现的，这样的控制精度不高，且温度波动较大。目前可采用室内远程传感器，通过变送器和各种调节器来控制冷热源管道上面的执行机构来实现温度自动调节的目的。

汽车行业应用先进、成熟的测试技术和制造技术，是成功开发性能优良、经济实用的汽车的基本原则。因此，计算机控制技术在汽车行业中得到了最为广泛的应用。例如，汽车性能测试系统中采用可编程序控制器 PLC 实现的控制系统。另外，分布式控制系统 DCS 成为了汽车过程工业自动化的主控系统，其特点是控制分散、操作显示集中，系统具有很高的可靠性和很强的功能。

在智能机器人技术中，利用各类传感器检测机器人自身或外部状态、信息，采用智能处理器进行信息、数据的分析和计算，控制其进行一定的运动、加工、探测等，也是计算机控制技术的典型应用。如图 1.4 所示为汽车生产线上喷漆的机器人。

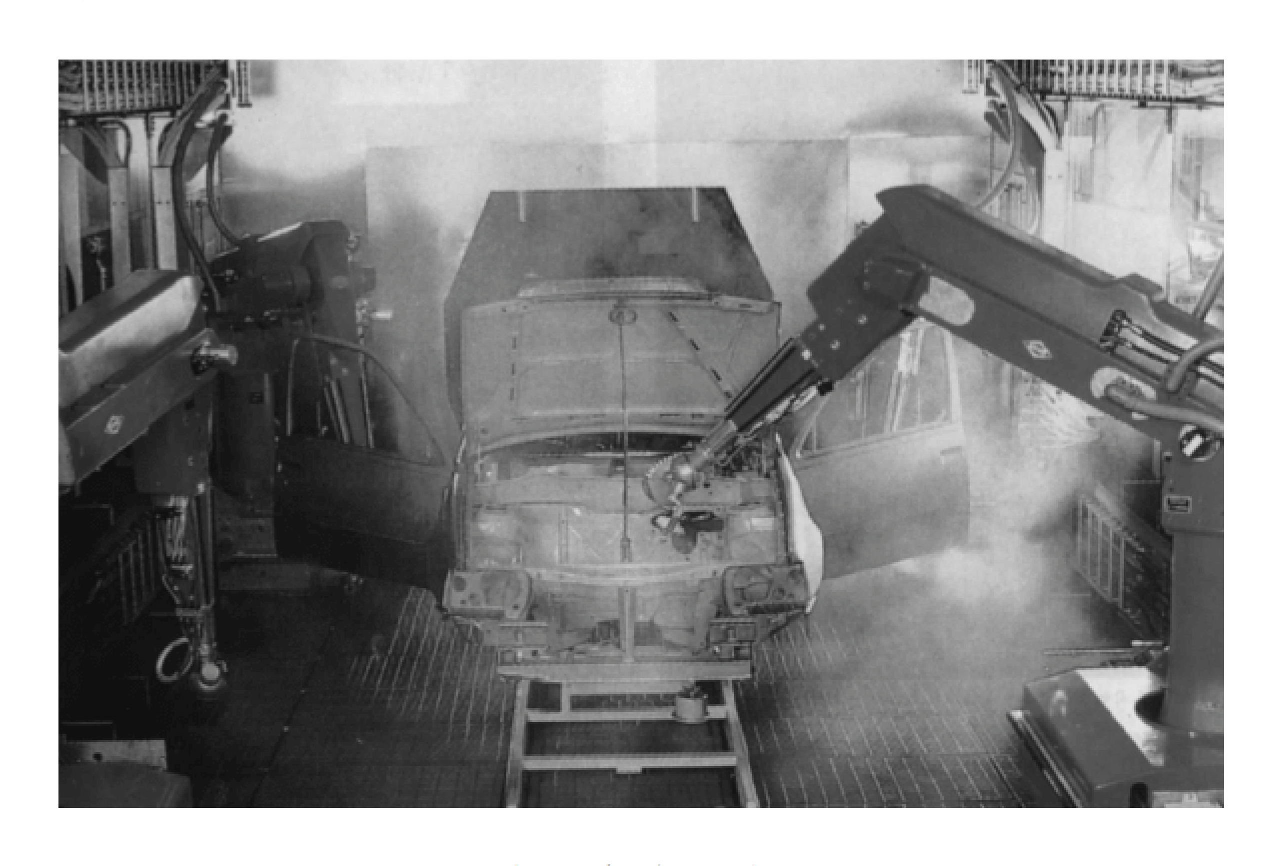


图 1.4　汽车生产线上喷漆的机器人

另外，航空航天领域中如火箭的发射、飞机的导航，智能楼宇自动控制系统中的门禁系统、电梯控制系统、火灾报警系统、安防报警系统，交通行业中的隧道 LED 照明控制系统中，都广泛采用了计算机控制技术。

1.2 计算机控制系统的典型形式

典型的工业生产过程分为三种：连续过程(Continuous Process)、离散过程(Discrete Process)和批量过程(Batch Process)。连续过程也称为流程工业，产品一般是流体，如液体、气体等；离散过程主要发生在制造业中，其产品是固态，按件计量，过程的输入输出变量为时间离散且幅度离散的量；批量过程为间歇性、多品种生产过程，其特点是连续过程和离散过程交替进行，如控制过程中配方的切换和生产工艺的改变是离散的，而配方和生产工艺确定后的生产过程是连续的。

控制一定生产过程的计算机控制系统所采取的结构形式有不同的方案，但方案的确定主要取决于生产过程的复杂程度，下面介绍几种典型的结构形式。

1.2.1 操作指导控制系统

操作指导控制系统(Operation Instruction Control System，OICS)不属于自动控制系统的范畴。计算机不直接用来控制生产过程，主要是对系统过程参数进行收集，并按照预定的算法计算各控制量，求出最佳值，通过显示或打印设备输出数据，以供操作人员进行参考。操作人员根据所获取的这些数据，对系统进行必要的操作与控制。

这种系统只对生产过程进行监视，不能进行自动控制。但通过计算机控制系统对生产过程或控制对象的大量参数进行巡回检测、处理、分析、记录、显示，可及时进行参数的超限报警，也可以对生产过程的各种趋势进行分析，有助于操作人员更好地对系统进行及时的控制。该系统具有结构简单、控制灵活、安全可靠等优点，尤其适用于未弄清控制规律的系统，是计算机应用于工业生产过程控制的最早期、最简单的一种系统。

1.2.2 直接数字控制系统

直接数字控制系统(Direct Digital Control System，DDCS)就是把数据采集得到的输入数据，根据预先确定的控制规律进行计算，然后输出控制量，通过执行机构直接控制生产过程，使被控参数保持在设定值，或是按照给定信号确定的变化规律进行变化的控制系统。系统结构图如图1.5所示。

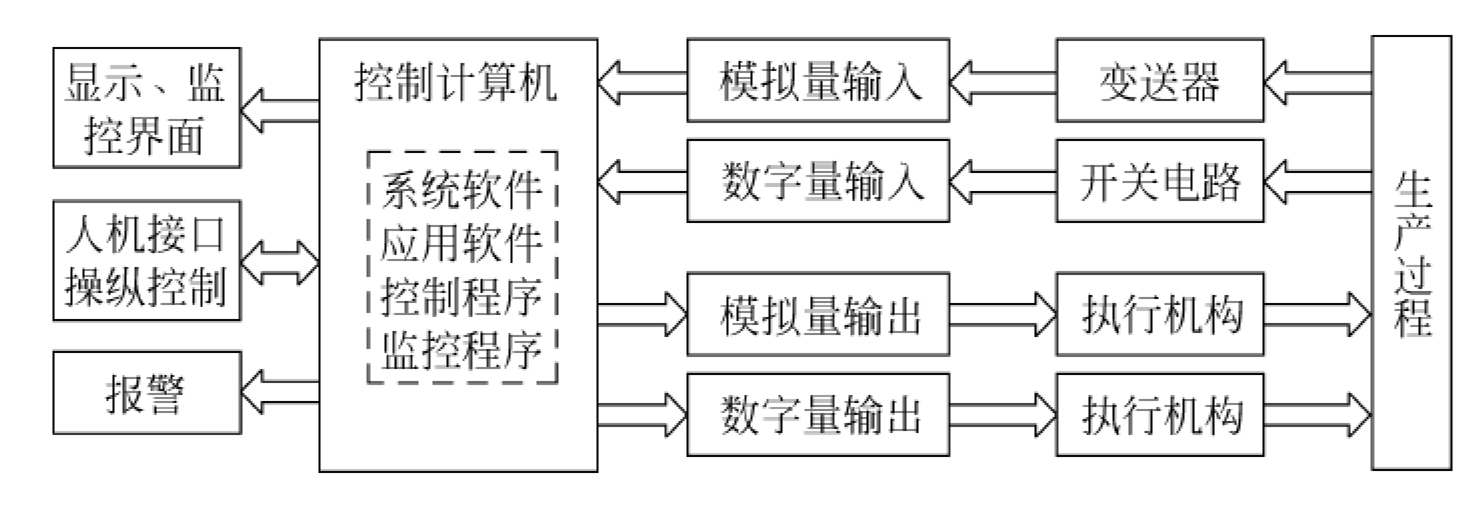


图1.5 直接数字控制系统结构图

直接数字控制系统与模拟控制系统类似，只是将模拟控制系统中的模拟控制器换成了计算机这种数字控制器。借助于计算机能够处理大量数据的优势，可以实现更加复杂控制规律的系统控制。

1.2.3 监督控制系统

监督控制系统(Supervisory Computer Control System,SCCS)是计算机根据原始工艺信息和其他的参数,按照描述生产过程的数学模型或其他方法,能够自动地改变模拟调节器或以直接数字控制方式工作的控制计算机中的给定值,从而使生产过程始终处于最优工况,保持生产的高质量、高效率、低消耗、低成本等。其结构图如图 1.6 所示。又称为设定值控制系统(Set Point Control System,SPCS)。

监督控制系统与直接数字控制系统相比,后者用于对生产过程进行控制的给定信号是预先设定的,或是由操作人员通过图 1.5 中的人机接口进行设定、修改,而不能由系统自动改动。而监督控制系统是 DDC 系统的改进,它采用两级控制方式,如图 1.6 所示,监督控制计算机(又称上位机)根据原始生产工艺数据和实时生产工况,可实时计算出被控系统的最优给定值,输出给控制计算机(或模拟调节器),由后者(也称下位机)实现系统的自动控制。同时,监督控制计算机也可以对生产状况进行分析,做出故障的诊断与预报。也就是说,监督控制计算机并不直接进行执行机构的控制,而是主要用于给定信号的监控。

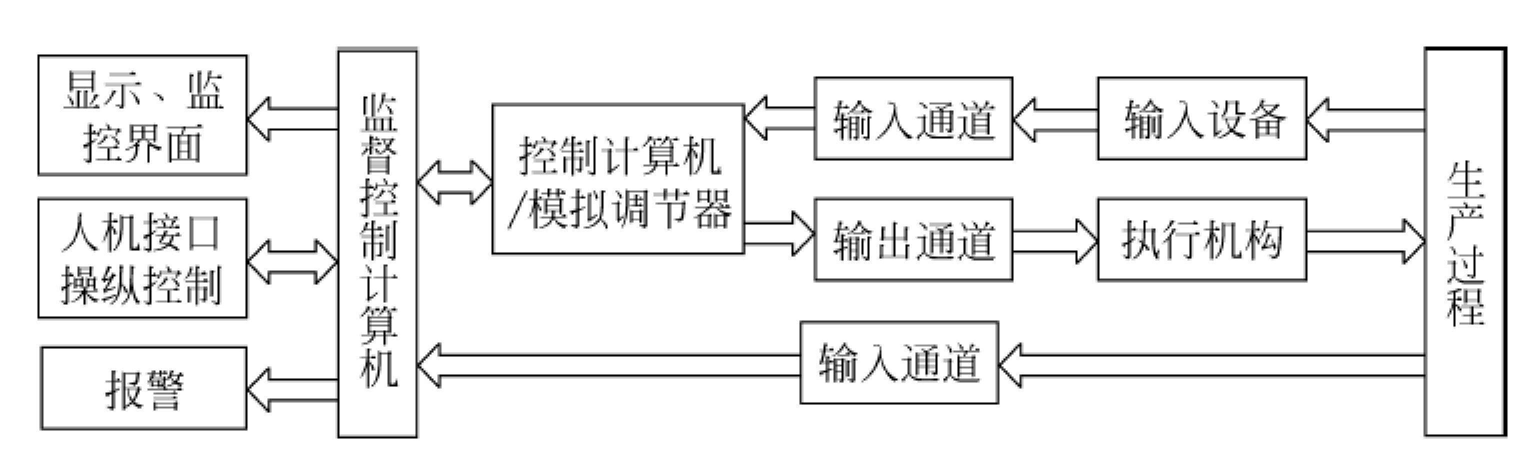


图 1.6 监督控制系统

1.2.4 集散控制系统

集散型计算机控制系统又名分布式计算机控制系统,简称集散控制系统(Distributed Control System,DCS),如图 1.7 所示。集散控制系统采用分散控制、集中操作、综合管理、分而自治的设计原则,把系统从下到上分为分散过程控制级、集中操作监控级、综合信息管

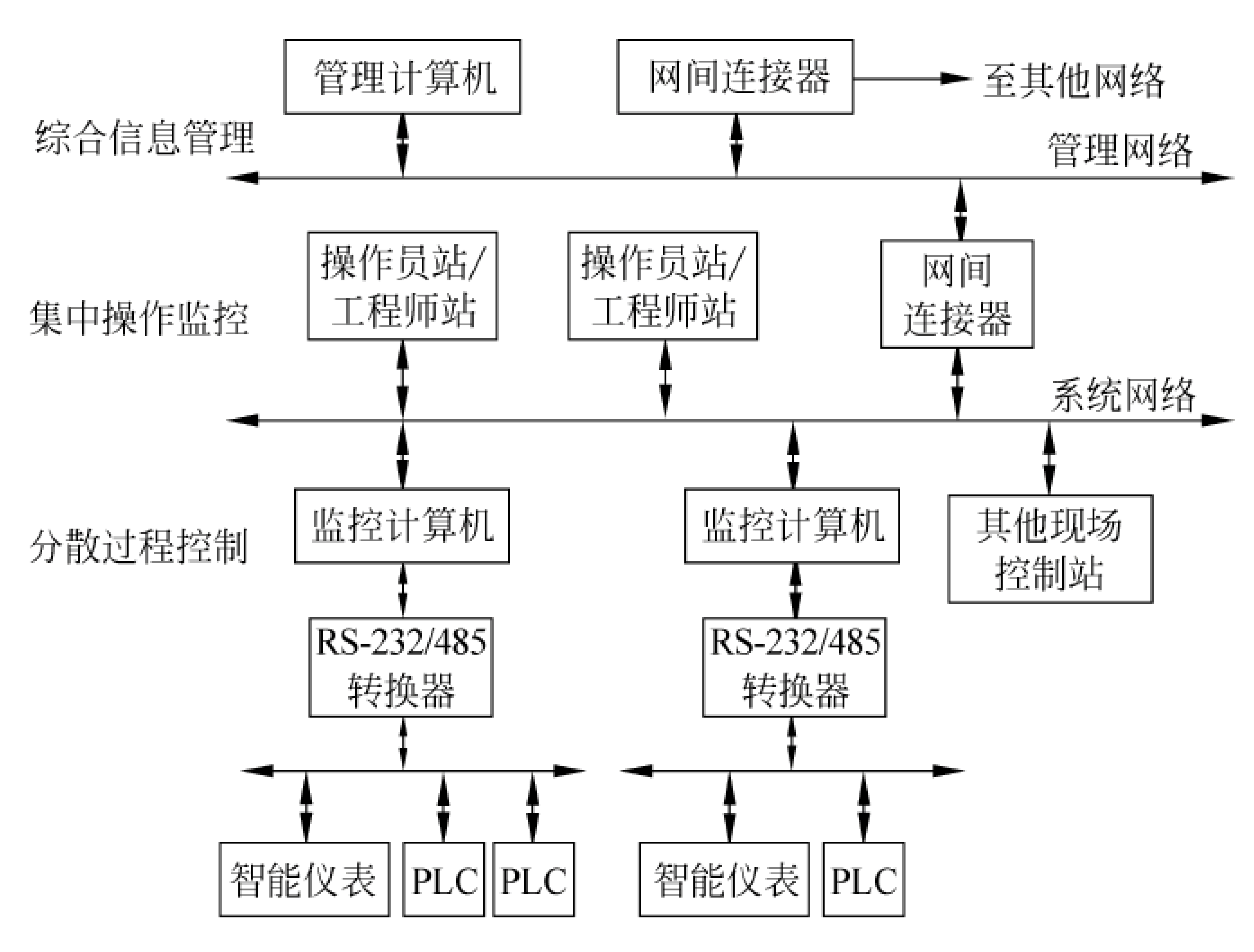


图 1.7 集散控制系统

理级,形成分级分布式控制结构。系统安全可靠,其通用灵活性能、最优控制性能和综合管理能力,为工业过程的计算机控制开创了新方法。集散控制系统综合了计算机技术、控制技术、通信技术、显示技术,即4C技术;集中了连续控制、批量控制、逻辑顺序控制、数据采集等功能。

先进的集散控制系统将以计算机集成制造系统为发展目标,以新的控制方法、现场总线、智能化仪表、专家系统、局域网络等新技术为支持,为用户实现与过程控制自动化相结合的管控一体化的综合集成系统。

1.2.5 现场总线控制系统

现场总线(Fieldbus)是将自动化系统最底层的现场控制器和现场智能仪表设备进行互联的实时控制、数字式、双向传输、多分支结构的通信网络。现场总线控制系统(Fieldbus Control System,FCS)是一种以现场总线为基础的分布式网络自动化系统,它既是现场通信网络系统,也是现场自动化系统,如图1.8所示。它是随着计算机技术、通信技术、计算机网络技术的发展而产生的工业自动化系统体系结构的变革,是在DCS的基础上发展起来的。FCS用一对通信线连接多台数字仪表取代一对信号线只能连接一台仪表的情况;用多变量、双向、数字通信方式取代单变量、单向、模拟传输方式;用多功能的现场数字仪表取代单功能的现场模拟仪表;废弃了DCS的输入/输出单元和控制站,由现场设备或现场仪表取而代之,即把DCS控制站的功能化整为零,分散地分配给现场仪表,从而构成虚拟控制站,实现彻底的分散控制。

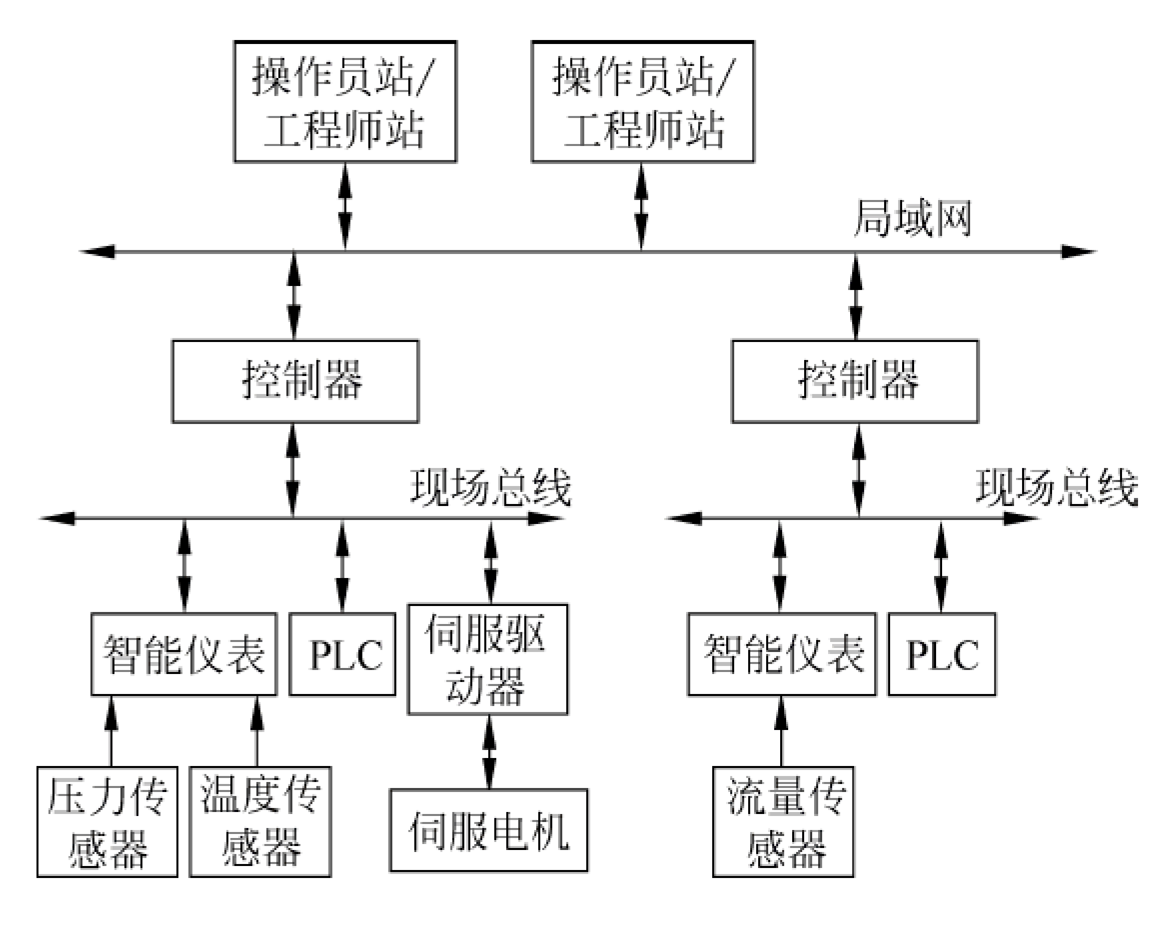


图1.8 现场总线控制系统

现场总线控制系统有两个显著的特点,一是信号传输实现了全数字化,二是实现了彻底分散。

过去10年内,世界上出现了多种现场总线的企业、集团或国家标准,目前较流行的现场总线主要有以下几种:CAN,LonWorks,Profibus,HART,FF等。现场总线控制技术在汽车行业应用的非常普遍,近两年国内新的汽车生产线和旧的生产线的改造,大部分都采用了现场总线的控制技术。

1.2.6　计算机集成制造系统

随着电子信息技术和自动化技术的发展，以及各种先进制造技术的进步，制造系统中许多以自动化为特征的单元技术得到了广泛的应用。如：计算机辅助设计(Computer Aided Design，CAD)、计算机辅助制造(Computer Aided Manufacturing，CAM)、计算机辅助工艺设计(Computer Aided Process Planning，CAPP)、计算机辅助工程(Computer Aided Engineering，CAE)、计算机辅助生产管理(Computer Aided Production Management，CAPM)、柔性制造系统(Flexible Manufacturing System，FMS)、柔性制造单元(Flexible Manufacturing Cell，FMC)等，为企业带来了显著的效益。但是，如果孤立地发展这些技术，必将会产生"自动化孤岛"现象，使其产生封闭性，相互之间不能很好地实现信息的传递与共享，必将会降低整体系统的工作效率。计算机集成制造系统(Computer Integrated Manufacturing System，CIMS)正是为了解决这样的问题而提出来的。

计算机集成制造(CIM)是一种企业组织、管理和生产的新思想，强调信息集成，借助于计算机软硬件，综合应用现代管理技术、制造技术、信息技术、自动化技术、系统技术，将企业生产全部过程中有关人、技术和经营管理三要素，以及信息流和物质流有机地进行集成，并优化运行，以实现产品生产的高效率、高质量和低成本，提高企业对市场变化的应变能力和综合竞争能力。CIMS 功能示意图如图 1.9 所示。

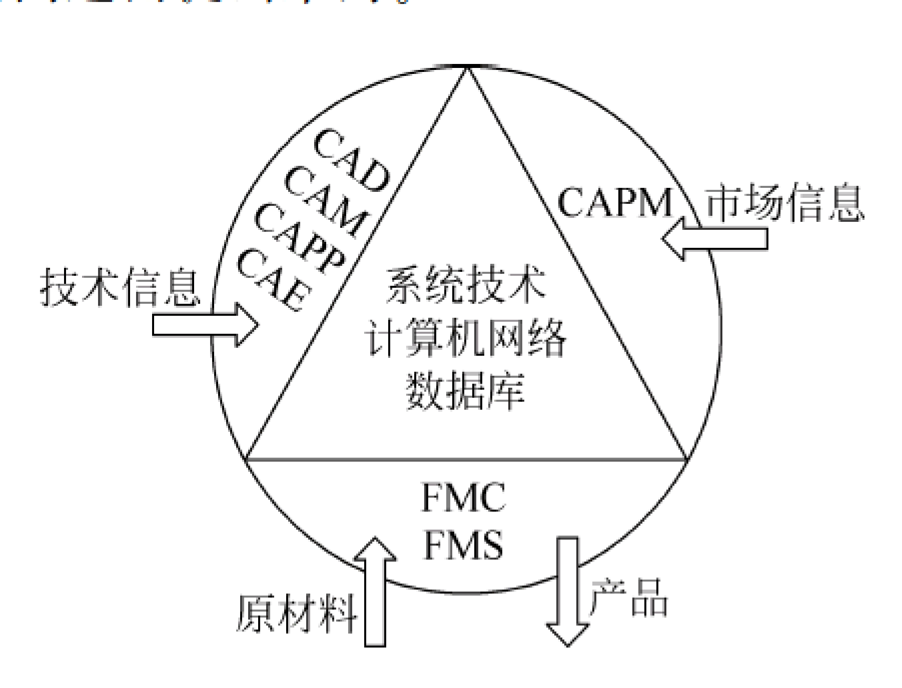


图 1.9　CIMS 功能示意图

CIMS 体系结构是用来描述研究对象整个系统的各个部分和各个方面的相互关系和层次结构，从系统理论角度将整个研究对象分为几个子系统，各个子系统相对独立自治、分布存在、并发运行和驱动。另外，CIMS 逻辑结构不仅体现了信息的传递与交换，更能反映出人在 CIMS 系统中的重要作用。因此可以说：人为因素是实施 CIMS 能否取得实际成效的关键。

1.3　计算机控制技术的发展

计算机控制技术是随着现代大型工业生产自动化的不断兴起而应运产生的综合控制系统，它紧密依赖于计算机技术、网络通信技术和控制技术的最新发展。

1.3.1　计算机控制技术的发展状况

自 1946 年世界上第一台电子计算机在美国问世以来，逐渐将自动控制技术推向了一个新的发展水平。使之经历了从"操作指导控制系统"到"计算机集成制造系统"的发展历程。

1. 操作指导控制系统阶段

1956 年计算机被用于工业过程控制，当时首先由 TRW(Thomson Ramo Woolrige)航

空公司与德克萨克(Texaco)公司提出方案,设计了一个用于炼油厂的聚合装置计算机控制系统。1959年第一台过程控制计算机系统在美国得克萨斯州的Port Arthur炼油厂正式投入运行。该系统可以控制26个流量信号、72个温度信号、3个压力信号,以及3个成分信号,主要用于数据处理和操作指导,属于早期的操作指导控制系统(OIC)。

2. 直接数字控制系统阶段

1962年英国的帝国化学工业公司(ICI)用一台计算机代替所有用于过程控制的模拟仪表,实现了244个数据量采集和129个阀门控制的直接数字控制系统(DDC)。

3. 集散控制系统阶段

1972年诞生了微型计算机,其速度快、体积小、可靠性高、价格便宜,被广泛应用于过程控制,使得计算机控制技术取得了突飞猛进的发展。1975年,世界上几个主要的计算机和仪表制造厂几乎同时生产出集散控制系统(DCS)。如美国霍尼韦尔(Honeywell)公司生产的TDC-2000。

4. 计算机集成制造系统阶段

20世纪80年代末推出了具有计算机辅助设计(CAD)、专家系统、控制和管理功能融为一体的新型集散控制系统。20世纪90年代,在流程工业中,计算机管控一体化系统从生产过程的全局出发,通过对生产活动所需的各种信息的集成,实现了常规的过程控制、先进控制、在线优化、生产调度、企业管理、经营决策等功能的统一管理,进入了计算机集成制造系统的发展阶段。

1.3.2 计算机控制技术的发展趋势

大规模及超大规模集成电路的发展,提高了计算机的可靠性和性价比,从而使计算机控制系统的应用也越来越广泛,并逐渐向智能化、网络化和集成化的方向发展。

智能化是指大力研究和发展智能控制系统,推广应用成熟的先进技术。使计算机控制系统能够无需人的干扰,就能自主地驱动执行设备实现其控制目标。

工业控制网络将向有线和无线相结合的方向发展。计算机网络技术、无线传输技术和智能传感器技术的结合,产生了基于无线技术的网络化智能传感器,使得工业现场的数据能够通过无线的方式传输和共享。

集成化一方面指各种专用的集成控制模块,如电机控制模块、调节阀控制模块、显示模块等,正在逐步取代计算机控制系统中的分立元件,这样能够更好地提高系统的可靠性。另一方面是指将企业内与生产有关的现代管理技术、制造技术、信息技术、自动化技术、系统技术,以及市场信息等进行有效的集成,并优化运行,以实现计算机控制系统在生产过程控制中更高效。

另外,工业控制软件已向组态化方向发展,如人机界面软件、控制软件、生产管理软件等,都有相应的成套应用软件供用户使用,为企业实现计算机集成制造(管控一体化)提供了比较完整的解决方案,使系统的设计更高效、便捷。

1.4 计算机控制技术的学习要点

计算机控制技术涵盖了计算机技术、接口技术、自动控制理论、运动控制、过程控制、检测技术及仪表、工程技术等相关基础知识,主要研究如何将计算机技术和自动控制理论应用于工业生产过程,并设计出所需要的计算机控制系统。

计算机控制技术课程是自动化等控制类专业的核心课程之一,学习的目的主要在于使学生了解和掌握以控制计算机为核心的控制系统的硬件、软件的基本理论、基本应用和基本设计方法。学习的过程中需要注意以下问题。

1. 注意区分几种控制系统

连续时间控制系统:系统中所有的信号在时间上是连续的,信号间的关系可以用微分方程来描述。

离散时间控制系统:系统中部分信号是以采样信号或数字信号的形式存在的。离散信号间的关系可用差分方程进行描述。

采样控制系统:系统内的控制器和对象均是连续信号处理器,用采样开关来达到多个对象共享一个控制器的目的。

数字控制系统:控制器只能处理数字(离散)信号,控制系统内必须具备转换器(如模/数转换、数/模转换)来完成连续信号与离散信号之间的相互转换。

2. 了解计算机控制技术涉及的相关理论

信号采样理论:采样定理等。

经典控制理论:频率相应法、根轨迹法、拉氏变换、传递函数、Z 变换、Bode 图等。

现代控制理论:状态空间法、系统辨识、最优控制、自适应控制、最佳滤波理论等。

智能控制理论:模糊控制、神经网络控制、专家系统、学习控制等。

3. 掌握一些硬件设计的相关知识

与硬件设计相关的主要知识包括:模拟电子技术、数字电子技术、接口技术、单片机技术、ARM 微处理器技术、DSP 技术、FPGA/CPLD 技术、总线技术、计算机应用技术、智能仪表应用技术、PLC 技术等。学习时,可以有重点的、带有选择性的掌握。

4. 掌握相关的一些应用软件

工业组态软件:组态王、力控、InTouch 软件、FIX 组态软件、WinCC 组态软件等。主要用于监控程序的设计。

PLC 编程软件:GE Cimplicity 软件、西门子 Step7 软件等,主要用于 PLC 的程序设计。

硬件仿真软件:Proteus、Multisim、Protel 等,可以用于硬件设计的过程中进行正确性验证,或分析运行结果。

硬件电路板设计软件:Protel、OrCAD、Proteus、PowerPCB 等,主要用于硬件电路板的设计与制版。

程序设计语言：C、Visual Basic、C++等，主要用于控制程序的编程。

程序设计软件：Keil C、VB、VC 等，用于控制程序的设计与调试。

控制策略分析、研究软件：MATLAB 等，主要用于控制算法的设计、分析与仿真。

5. 掌握一些控制算法

计算机控制系统常用的控制算法包括：基本 PID、改进 PID、模糊 PID、最小拍控制、Smith 预估、达林算法、多变量解耦控制、自适应算法、神经网络算法等。可以从最基本的进行学习和掌握，由简单到复杂。

6. 掌握一些常用传感器的应用

针对工业控制当中常常需要测量、监控的一些物理量，掌握一些常用的传感器（或变送器）的应用，如温度、压力、流量等相关的传感器。

当然，要完全掌握以上知识，并不是学习一两门课程就可以做到的，也不是短短一个学期就可以达到的，需要在学习“计算机控制技术”的同时，不仅要多学、多练，同时要广泛与本专业的其他课程相联系，通过网络、参考书或专业期刊等搜集资料，做到融会贯通。另外，如果能通过实际动手搭建一个系统，并进行编程、调试，将会得到较大的收效。如果硬件条件不能满足，最好利用一些仿真软件，通过设计、仿真的过程，可以达到学习、应用的效果。

习题

1. 什么是计算机控制系统？它主要有哪几部分组成？各部分的作用是什么？
2. 简述计算机控制系统的控制过程。
3. 计算机控制系统的典型形式有几种？各有什么特点？
4. 试列举计算机控制系统的最新发展趋势，说明研究计算机控制系统的重要性。
5. 列举计算机控制技术的相关理论，系统设计时需考虑什么问题？
6. 计算机控制系统的发展趋势如何？

第2章 计算机控制系统范例

为了使初学者对计算机控制系统有一个初步的了解和整体的认识，本章以一个温度控制系统为例，简单介绍一下系统的硬件、软件结构，以及系统设计的基本方法。

在各种计算机控制系统中，PC板卡式控制是最基本的构成形式，它充分利用了计算机的主板、总线、电源、人机界面、软件资源等，使设计方便、快捷，便于入门学习。因此，本章所介绍系统是应用研华公司的多功能数据采集卡PCL-818L(模拟量输入)和康拓公司32路光隔开关量功放输出卡IPC5373来完成带夹套锅炉的温度信号的采集与控制。要求设计一套完整的计算机温度测控系统，可以实时检测当前温度，并进行温度的实时控制，以达到所期望的温度值，测温范围为0～100℃，控制精度为±0.5℃。

2.1 温度控制系统硬件结构

根据设计要求，构建系统硬件结构框图如图2.1所示。这里，采用铂热电阻Pt100温度传感器进行温度测量；选择AI-808A人工智能工业调节器作为温度变送器；通过研华的接线端子PCLD8115将测量的模拟信号接入数据采集卡PCL-818L进行A/D转换，然后输入研华工控机IPC-610H；处理后的数字量通过康拓开关量功放输出卡IPC5373驱动低压继电器；继电器带动交流接触器，实现电加热丝或是冷却泵的通电与断电控制，从而实现对被控对象(带夹套锅炉)的温度进行升温和降温处理。下面对每一部分硬件作简要介绍。

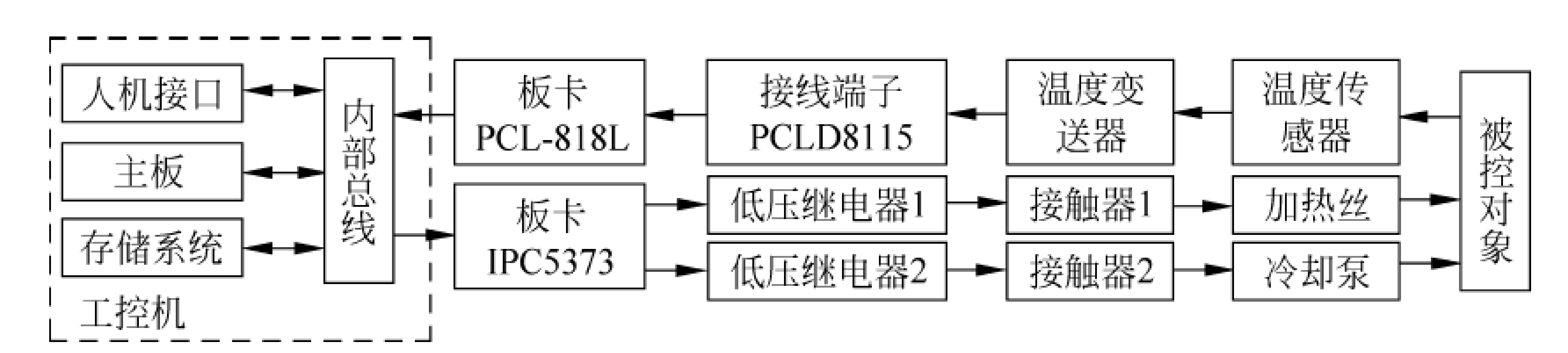


图2.1 温度控制系统硬件结构

2.1.1 控制计算机

控制计算机是整个计算机控制系统的核心。这里选用的是研华IPC-610H工控机(见图2.2)及配套设备，其主要特点是：4U高，支持14槽背板；前置USB/PS2接口；能抗冲击、抗振荡，并且能在高温下稳定工作；支持ATX母板和400W PFC电源；它通过插入总线插槽的模拟输入板卡输入被控对



图2.2 IPC-610H工控机

象的检测参数,按照应用程序自动地进行信息处理、分析和计算,并做出相应的控制决策或调节,通过插入总线插槽的输出板卡及时发出控制命令,实现对象的实时控制。

2.1.2 传感器

传感器选用铂热电阻 Pt100 温度传感器,0℃时标称值为 100Ω,它的阻值会随着温度上升而呈近似匀速的增长,100℃时标称值约为 138.5Ω。热电阻的电阻值和温度一般可以用以下的近似关系式表示:

$$R_t = R_{t0}[1+\alpha(t-t_0)] \tag{2.1}$$

式中,R_t 为温度 t 时对应的阻值;R_{t0} 为温度 t_0(通常 $t_0=0$℃)时对应电阻值;α 为温度系数。

热电阻采用三线制接法,如图 2.3(a)所示。采用三线制是为了消除连接导线电阻引起的测量误差。这是因为测量热电阻的电路一般是不平衡电桥。热电阻作为电桥的一个桥臂电阻,其连接导线(从热电阻到中控室)也成为桥臂电阻的一部分,这一部分电阻是未知的且随环境温度变化,造成测量误差。采用三线制,将导线一根接到电桥的电源端,其余两根分别接到热电阻所在的桥臂及与其相邻的桥臂上,这样消除了导线线路电阻带来的测量误差。

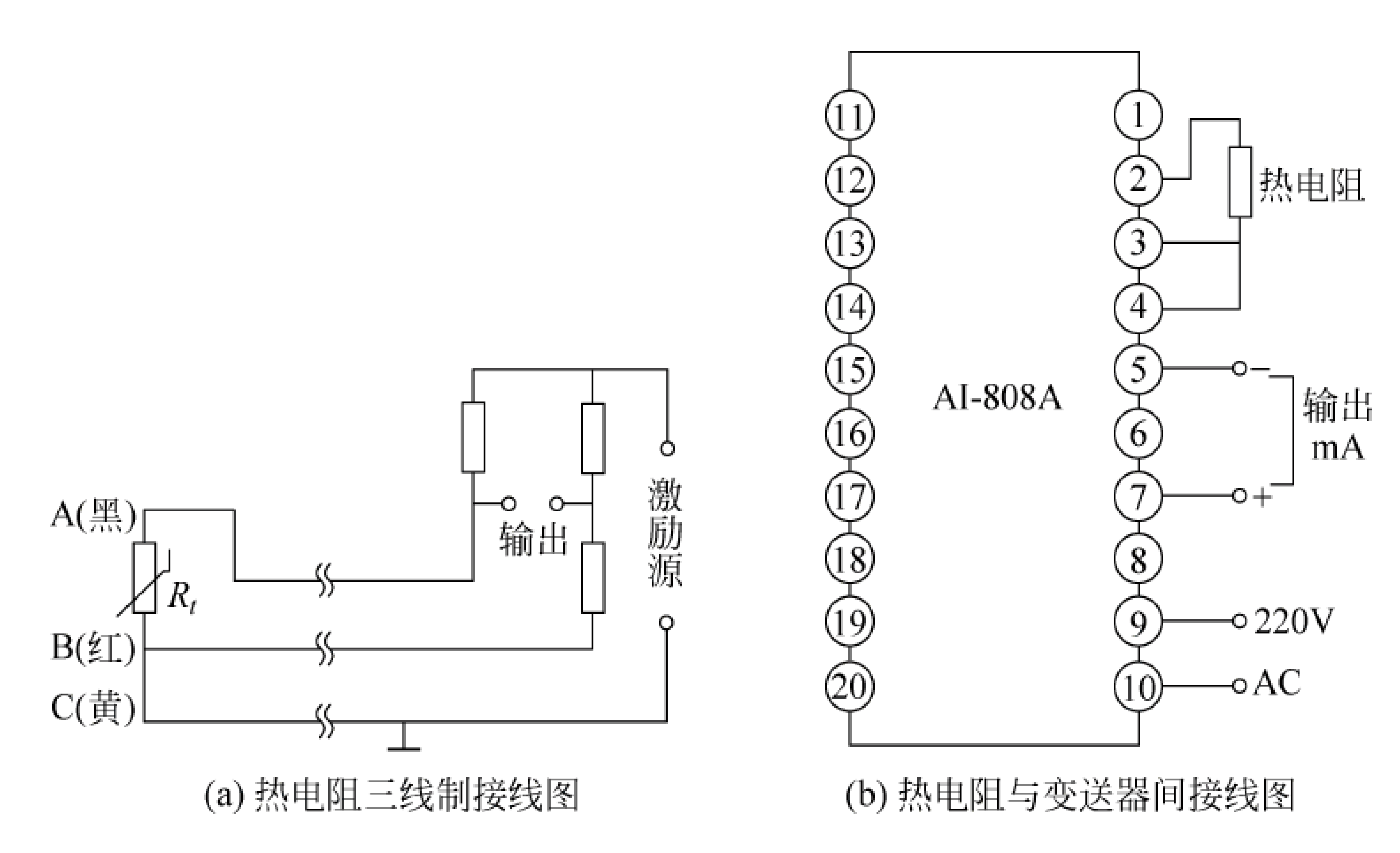


图 2.3 热电阻接线图

2.1.3 温度变送器

选择 AI-808A 人工智能工业调节器作为与 Pt100 相对应的温度变送器,具体接线如图 2.3(b)所示,将 Pt100 热电阻的 A、B、C 三端分别接至 AI-808A 的 2、3、4 接线端子,则根据 0~100℃的环境温度测得的 Pt100 的电阻值输入变送器,输出 4~20mA 的标准电流信号(从 AI-808A 的 7(+)、5(−)端输出);在输出端并联 250Ω 电阻后,得到 1~5V 的标准电压信号即可通过 PCLD8115 接线端子输入 PCL-818L 板卡。

其中人工智能工业调节器 AI-808A 的参数设置如下。

- HIAL(测量上限):100。
- LoAL(测量下限):0。

- dHAL(正偏差报警限值)：5.0。
- dLAL(负偏差报警限值)：5.0。
- Ctrl(控制方式采用位式调节)：0。
- Sn(用于选择输入规格，此时选定输入为Pt100)：21。
- DIP(显示格式为000.0，小数点在十位)：1。
- DIL(输入下限显示值)：0。
- dIH(输入上限显示值)：100。
- OP1(选择4～20mA线性电流输出)：4。
- Addr(输出下限为4mA)：40。
- bAud(定义变送输出电流上限为20mA)：200。
- run(确定仪表为自动调节状态)：1。
- Loc(仪表选择为可设置全部参数及给定)：808。

2.1.4　输入输出板卡

此系统用到两种输入输出板卡，分别是：研华公司的多功能数据采集卡PCL-818L和康拓公司32路光隔开关量功放输出卡IPC5373。

1. 多功能数据采集卡PCL-818L

PCL-818L是PCL-818系列中的入门级板卡，是为IBM或其他兼容计算机设计的基于ISA总线的一款高性能、高速多功能数据采集卡。主要具有以下特点。

- 16路单端或8路差分模拟量输入。
- 12位A/D转换器，带DMA的自动通道增益扫描。
- 每个输入通道的增益可编程，包括×0.5，×1，×2，×4，×8。
- 板上带有一个1K的采样FIFO(先入先出)缓冲器和可编程中断。
- 16路数字量输入及16路数字量输出。
- 软件可选择模拟量输入范围，双极性：±0.625V，±1.25V，±2.5V，±5V，±10V；单极性：0～1.25V，0～2.5V，0～5V，0～10V。
- 2通道12位多路切换模拟输出，1通道16位通用定时/计数器。

使用前须对照"研华PCL-818系列板卡中文手册"对板卡进行开关和跳线的设置。然后根据使用手册的引脚说明进行连线。PCL-818L板卡实物图如图2.4所示。



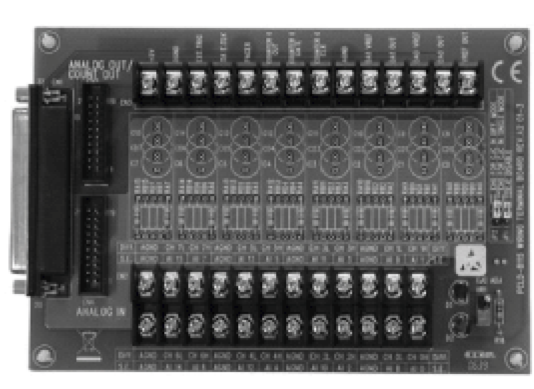
图2.4　PCL-818L板卡实物图

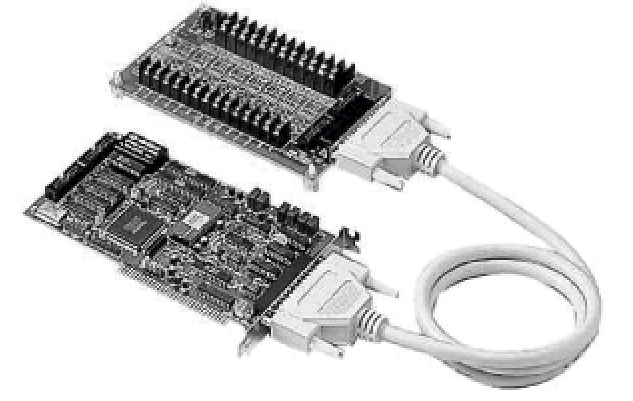
要使PCL-818L正常使用，还需进行如下准备工作：

- 在工控机上安装Device Manager软件和32bDLL驱动。
- 使用Device Manager配置软件对板卡进行配置。
- 使用Device Manager配置软件对板卡进行硬件测试。

- 进行板卡的正确接线。
- 然后使用组态王或其他组态软件，或 VB 等开发软件实现通信、数据采集和控制等。

详细请认真阅读 PCL-818L 用户使用手册。另外，此板卡插入计算机总线后，为方便外接信号线，通过研华的接线端子 PCLD8115 将测量的模拟信号接入数据采集卡 PCL-818L，PCLD8115 接线端子及其与 PCL-818L 连接图如图 2.5 所示。





(a) PCLD8115接线端子 (b) 接线端子及与PCL-818L连接图

图 2.5 PCLD8115 接线端子及其与 PCL-818L 连接图

2. 光隔开关量功放输出板 IPC5373

光隔开关量功放输出板 IPC5373 是一块通用光电隔离型的开关量输出板，如图 2.6 所示。它包括 32 路开关量输出，可以实现 PC 总线与被测工业设备或数字仪器之间完全的电隔离，隔离电压 2500Vrms，以消除公共地线和电源的干扰，从而使工业设备和微机系统可靠工作；此外它还具有较强的输出驱动能力和电平转换能力，采用达林顿管功放集电极开路输出，负载电源 5～40V，驱动电流单路最大 200mA，每片达林顿管负载电流最大 500mA，可直接驱动继电器、电磁阀等；占用 4 个连续口地址：154H～157H；各输出信号具有锁存功能，上电复位清零；用户接口为 40 芯扁平电缆插座。

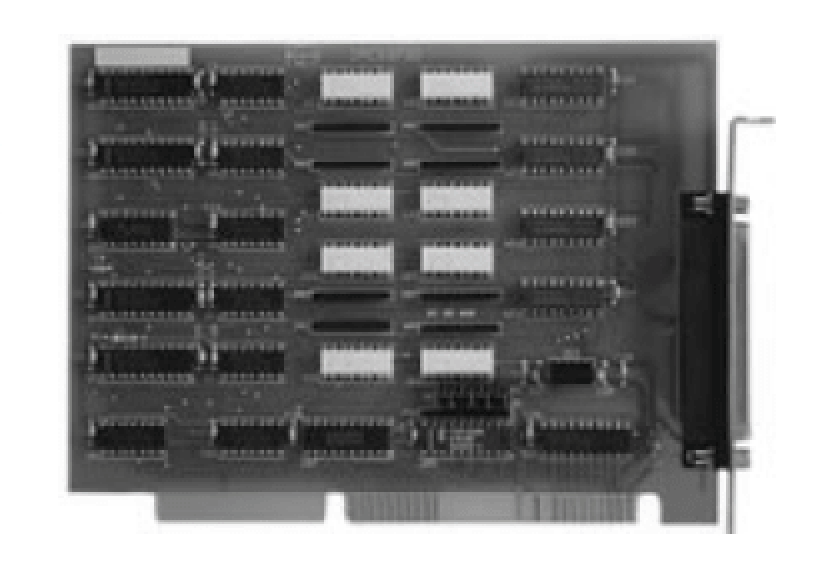


图 2.6 IPC5373 实物图

IPC5373 板和 PCL-818L 一样也是作为 PC 总线板插架系统的一部分，直接插在 PC 总线的任意一个槽中。具体接线及使用方法请认真阅读 IPC5373 用户使用说明书。

2.1.5 执行机构

执行机构的作用是接受计算机发出的控制信号，并把它转换成执行机构的动作，使被控对象按预先规定的要求进行调整，保证其正常运行，即控制生产过程。常用的执行机构有各种电动、液动及气动开关，电液伺服阀，交直流电动机，步进电机，各种有触点和无触点开关、电磁阀等。在系统设计中需根据系统的要求来选择。

本系统中主要根据当前的温度对被控对象进行加热或降温处理，所以执行机构主要使用了两套低压继电器（24V）和交流接触器（220V），用来接通或断开加热丝及冷却泵的供电电源，从而实现加热和降温处理。

2.1.6　其他外围设备

外围设备主要是为了辅助工控机主机的功能而配置的，用来显示、存储、打印、记录各种数据，包括输入设备、输出设备和存储设备。常用的外围设备有打印机、记录仪、显示器、外部存储器(硬盘、光盘等)、记录仪、声光报警器等。

除了上述硬件以外，还有加热丝、冷却泵、导线、电源等设备。根据需要也可以增加网络通信接口实现与其他设备的网络通信等。

2.2　温度控制系统软件设计

为了实时进行温度的采集与监控，必须通过软件的设计使工控机和数据采集硬件形成一个完整的数据采集、分析、显示和控制的体系。即控制系统的硬件电路确定之后，其主要功能将依赖于软件来实现。

控制软件的编写常采用以下三种开发工具：一是采用机器语言、汇编语言等面向机器的低级语言来编写；二是采用 C、Visual Basic、C++ 等高级语言来编写；三是采用监控组态软件来编写。为使初学者便于接受，这里采用较为直观，也相对较易实现的组态软件来进行控制软件的设计，组态软件选用亚控公司的工业组态软件组态王(Kingview)。

组态王是目前国内具有自主知识产权、市场占有率相对较高的组态软件。组态王运行于 Microsoft Windows 9x/NT/XP 平台。成功地应用于我国工业领域的很多行业，如石油、化工、电力、冶金、造纸、橡胶、环保、机械制造等。在设计过程中主要有以下特点。

(1) 用抽象的图形画面来模拟实际的工业现场和相应的工控设备。

(2) 画面上的图素用动画的模式来模拟现场设备的运行情况，并且让操作者可直观的在相关图素中输入控制设备的参数或指令来进行系统的控制。

(3) 通过创建一个具体的数据库，并用此数据库中的变量描述工控对象的各种属性，如水位、流量、温度等，并和画面图素的动画相关联。

详细的设计方法与步骤在第 9 章将进一步介绍。

2.2.1　系统的图形画面

运行已安装好的组态王程序，在工程管理器中创建一个新工程；在工程浏览器左侧树形菜单中选择新建一个画面；打开图库选择相关图素，构建系统画面，如图 2.7 所示。

2.2.2　板卡设备定义

为了让输入、输出信号能够与上述画面当中的图素相关联，需要进行设备定义，也就是所用到的输入、输出板卡定义。在组态王工程浏览器的左侧选择“设备”中的“板卡”，在右侧双击“新建”，运行“设备配置向导”，对 PCL-818L 和 IPC5373 两个板卡进行设置：PCL-818L 地址设置为“300”，详细说明参见研华 PCL-818 系列板卡中文手册，设置对话框如图 2.8 所示；IPC5373 地址设置为“154”，详细设置请参考 IPC5373 使用说明书，设置对话框如图 2.9 所示。

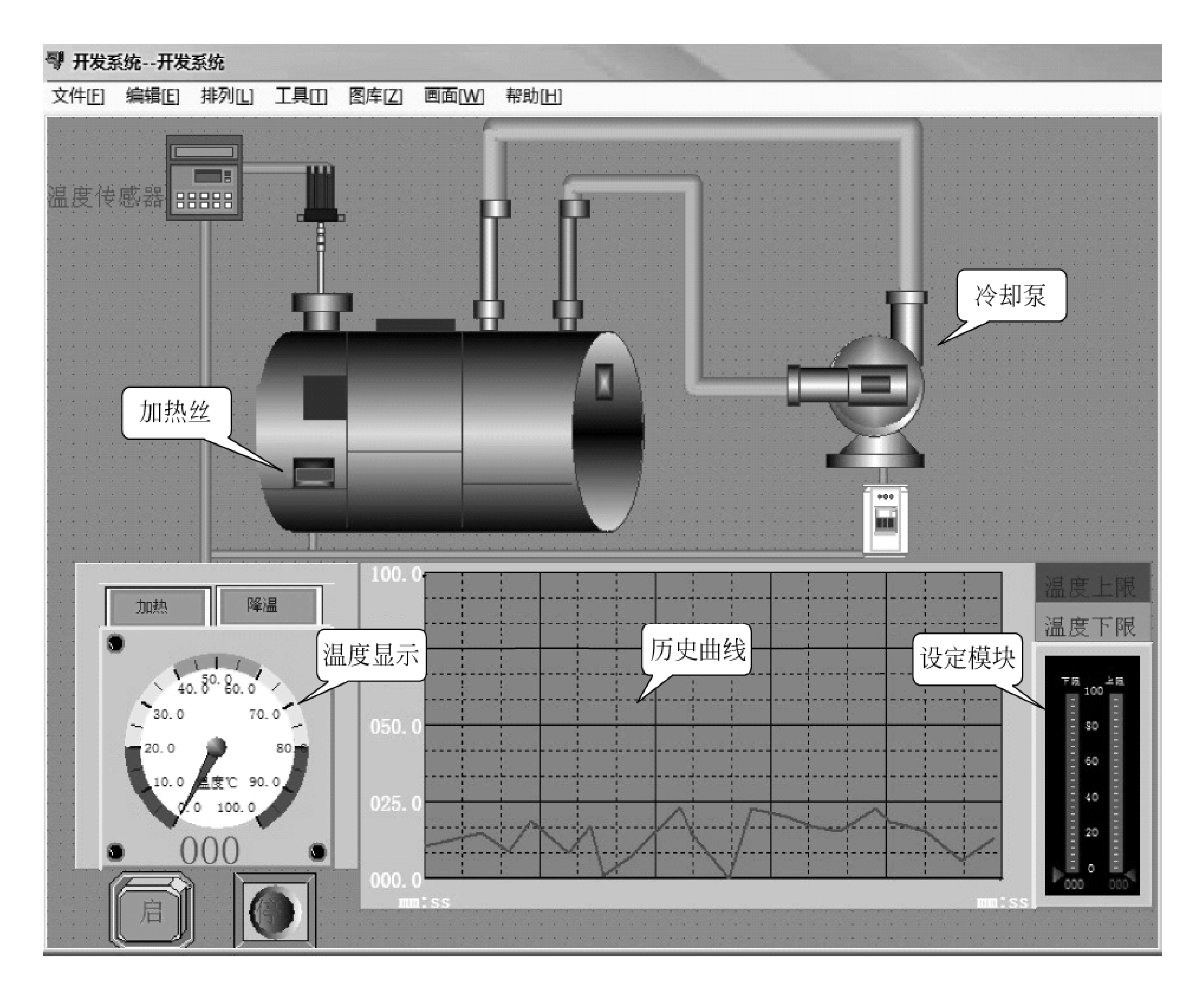


图 2.7　系统图形画面



图 2.8　输入板卡 PCL-818L 地址设置



图 2.9　输出板卡 IPC5373 地址设置

2.2.3　系统变量定义

系统所用变量如图 2.10 所示。主要是三类变量：一个模拟量输入 I/O 变量、一个数字量输出 I/O 变量和若干内存变量。

变量名	变量描述	变量类型	ID	连接设备	寄存器
中间温度		内存整型	35		
运行、停止指令		内存离散	36		
闪烁变量		内存整型	27		
冷却信号		内存整型	28		
冷却显示		内存离散	30		
控制温度下限		内存整型	34		
控制温度上限		内存整型	33		
加热信号		内存整型	25		
加热显示		内存离散	29		
加热、冷却输出		I/O整型	23	新IO设备1	DO0
断续冷却信号		内存整型	31		
断续加热信号		内存整型	26		
断续变量		内存整型	32		
time0		内存实型	22		
AI1		I/O实型	21	新IO设备	AD1.G1

图 2.10 系统所用变量

1. 模拟量输入 I/O 变量

模拟量输入的传感器采用的是 Pt100，温度测量范围取 0～100℃，经变送器 AI-808A 线性输出 4～20mA 电流，经 250Ω 电阻将电流信号转换为 1～5V 电压，并接入板卡 PCL-818L。因此，输入变量定义为“AI1”，变量类型选“I/O 实数”，最小值设为“0”，最大值设为“100”，最小原始值设为“2458”(对应 0℃)，最大原始值设为“4095”(对应 100℃)，连接设备为 PCL-818L(新 IO 设备)，寄存器选为“AD1. G1”，数据类型选“SHORT”，采样频率为“500”，读写属性为“只读”，如图 2.11 所示。



图 2.11 模拟量输入 I/O 变量定义

2. 数字量输出 I/O 变量

数字量输出变量通过 IPC5373 输出，用于驱动外部低压继电器(24V)和交流接触器(220V)。IPC5373 板卡有 4 个寄存器输出，每个寄存器有 8 个通道(0～7)，设定值为 154H 时，使用 154 寄存器的 8 个通道。各通道的导通与断开由二进制量控制，为 1 则导通，为 0

则断开，因此输出值为0～255。将其定义为“加热、冷却输出”，变量类型选“I/O整数”，最小值设为“0”，最大值设为“255”，最小原始值设为“0”，最大原始值设为“255”，连接设备为IPC5373(新IO设备1)，寄存器选为“D00”，数据类型选“BYTE”，采样频率为“1000”，读写属性为“只写”，如图2.12所示。

图2.12 数字量输出I/O变量定义

3. 内存变量

其他内存变量是为了配合数据的处理与控制，例如控制相关指示灯的当前状态，像加热显示、冷却显示、加热信号、冷却信号、断续加热信号、断续冷却信号；中间变量也可以控制系统各状态的运行时间，像闪烁变量、断续变量；还可设定系统运行的设定值与控制按键值，像控制温度下限、控制温度上限、中间温度、运行与停止指令等；可从图2.10中看到，这里不再详述。

2.2.4 动画连接与程序代码的编写

建立图2.7画面中各个对象的动画连接，将其与所定义的相关变量连接起来(具体可参考第9章组态软件应用部分)，然后进行程序代码的编写。通过组态王工程浏览器的左侧选择“应用程序命令语言”，并写入程序代码，用来描述整个系统的控制情况，参考代码如下：

```
if(运行、停止指令 = = 1)
{
中间温度 = (控制温度上限 + 控制温度下限)/2;
//************** 连续冷却、加热 ****************
if(\\本站点\AI1 <控制温度下限)
{
冷却显示 = 0;
加热信号 = 1;
time0 = time0 + 1;
}
if(\\本站点\AI1 >控制温度上限)
{
```

```
加热显示 = 0;
冷却信号 = 1;
time0 = time0 + 1;
}
if((time0 = = 5)&&(\\本站点\AI1 <控制温度下限))
{
加热显示 = 1;
加热、冷却输出 = 64;
time0 = 0;
断续变量 = 0;
闪烁变量 = 0;
}
if((time0 = = 5)&&(\\本站点\AI1 >控制温度上限))
{
冷却显示 = 1;
加热、冷却输出 = 32;
time0 = 0;
断续变量 = 0;
闪烁变量 = 0;
}
if((控制温度下限<= \\本站点\AI1)&&(\\本站点\AI1 <= 控制温度上限))
//************* 断续加热、冷却 *************
{
断续变量 = 断续变量 + 1;
闪烁变量 = 闪烁变量 + 1;
    if(加热信号 = = 1)
{
断续加热信号 = 1;
}
    if(冷却信号 = = 1)
{
断续冷却信号 = 1;
}
    if((断续变量 = = 20)&&(断续加热信号 = = 1)&&(加热、冷却输出 = = 64))
{
加热、冷却输出 = 0;
断续变量 = 0;
}
    if((断续变量 = = 20)&&(断续加热信号 = = 1)&&(加热、冷却输出 = = 0))
{
加热、冷却输出 = 64;
断续变量 = 0;
}
    if((断续变量 = = 20)&&(断续冷却信号 = = 1)&&(加热、冷却输出 = = 32))
{
加热、冷却输出 = 0;
断续变量 = 0;
}
    if((断续变量 = = 20)&&(断续冷却信号 = = 1)&&(加热、冷却输出 = = 0))
{
加热、冷却输出 = 32;
```

```
断续变量 = 0;
}
    if(\\本站点\AI1 <= 中间温度)
{
冷却显示 = 0;
断续冷却信号 = 0;
冷却信号 = 0;
}
    if(\\本站点\AI1 >中间温度)
{
加热显示 = 0;
断续加热信号 = 0;
加热信号 = 0;
}
}
if((断续加热信号 = = 1)&&(闪烁变量 = = 2)&&(\\本站点\AI1 <中间温度)&&(加热显示 = = 1))
//************** 断续闪烁 **************
{
加热显示 = 0;
闪烁变量 = 0;
}
if((断续加热信号 = = 1)&&(闪烁变量 = = 2)&&(\\本站点\AI1 <中间温度)&&(加热显示 = = 0))
{
加热显示 = 1;
闪烁变量 = 0;
}
if((断续冷却信号 = = 1)&&(闪烁变量 = = 2)&&(\\本站点\AI1 >中间温度)&&(冷却显示 = = 1))
{
冷却显示 = 0;
闪烁变量 = 0;
}
if((断续冷却信号 = = 1)&&(闪烁变量 = = 2)&&(\\本站点\AI1 >中间温度)&&(冷却显示 = = 0))
{
冷却显示 = 1;
闪烁变量 = 0;
}
}
else
    加热、冷却输出 = 0;}
```

2.3 系统工作过程

下面进行实验。夹套锅炉中水的当前温度经 Pt100 测量后送到温度变送器 AI-808A，经变送器及 250Ω 电阻转换后得到 1～5V 的模拟电压信号，由 PCL-818L 板卡进行 A/D 转换，转换为 12 位二进制数(2458～4096)输入到工控机。主控计算机安装组态软件组态王 6.53，根据输入的数据进行控制运算，运算输出的两路(一路用于控制加热丝，一路用于控制冷却泵)数字量结果由输出板卡 IPC5373 进行光隔和功率放大，然后分别输出到两路低压

继电器(24V),控制低压继电器常开触点的关断,低压继电器的常开触点作为两路接触器(220V)线圈电流的控制开关,当低压继电器常开触点闭合时,接触器线圈得电,接触器常开触点闭合,分别控制加热器或冷却泵工作。

先做如下设定:设定温度为 T_0,根据允许误差为±0.5℃,则温度控制上限为 $T_1=T_0+0.5$℃,下限为 $T_2=T_0-0.5$℃,则工作过程可描述如下。

(1) 若水温低于 T_2,经检测换算后输入到组态王监控界面,运算后 IPC5373 设定值为 64(0x40),输出通道 1 导通,低压继电器 1 线圈得 24V 电压,常开触点闭合,使得接触器 1 线圈得 220V 电压,接触器 1 的常开触点闭合,加热器通电,开始连续工作,此时系统处于连续加热状态。

当温度上升到 T_2 后,系统退出连续加热状态,进入断续加热状态,加热器工作 2s,停止工作 2s,依次循环,直到水温升到中间温度 T_0,加热控制过程结束,令 IPC5373 设定值为 0(0x00),输出通道全部断开,加热器、冷却泵都不工作,组态王继续监控水温,直到水温超过 T_1,或水温低于 T_2,系统进入相应控制过程。

(2) 若水温高于 T_1,经检测换算后输入到组态王监控界面,运算后 IPC5373 设定值为 32(0x20),输出通道 2 导通,低压继电器 2 线圈得 24V 电压,常开触点闭合,使得接触器 2 线圈得 220V 电压,接触器 2 的常开触点闭合,冷却泵开始连续工作,此时系统处于连续冷却状态。

当温度下降到 T_1 后,系统退出连续冷却状态,进入断续冷却状态,即冷却泵工作 2s 后停止工作 2s,依次循环,直到水温降到中间温度 T_0,冷却控制过程结束,令 IPC5373 设定值为 0(0x00),输出通道全部断开,加热器、冷却泵都不工作,组态王继续监控水温,直到水温高于 T_1,或水温低于 T_2,系统进入相应控制过程。

(3) 若水温处于 T_1 和 T_2 之间,IPC5373 设定值为 0(0x00),输出通道全部断开,加热器、冷却泵都不工作,组态王继续监控水温,直到水温超过 T_1,或水温低于 T_2,系统进入相应控制过程。

习题

1. 请根据 2.2.4 节代码画出程序流程图。
2. 根据本章实例,请说出计算机控制系统基本硬件及软件结构。
3. 结合你所了解的内容,举出一个计算机控制系统的应用实例。

第3章 硬件设计基础

计算机控制系统是利用主控计算机作为系统的主控制器,来实现工业生产过程自动控制的系统,如图 3.1 所示。计算机控制系统基本硬件结构主要包括主控机、人机接口、模拟量输入通道、模拟量输出通道、数字量输入通道、数字量输出通道、测量变送环节、执行机构、输入电气开关、输出电气开关、被控制对象,以及其他特殊功能扩展板卡等。因此,计算机控制系统的硬件设计主要包括以下几项。

(1) 主控机的选型;

(2) 输入/输出过程通道的设计;

(3) 系统各类总线形式的选择;

(4) 特殊功能板卡的设计;

(5) 执行机构的选择;

(6) 测量变送环节的选择;

(7) 接口电路的设计;

(8) 系统的抗干扰技术设计等。

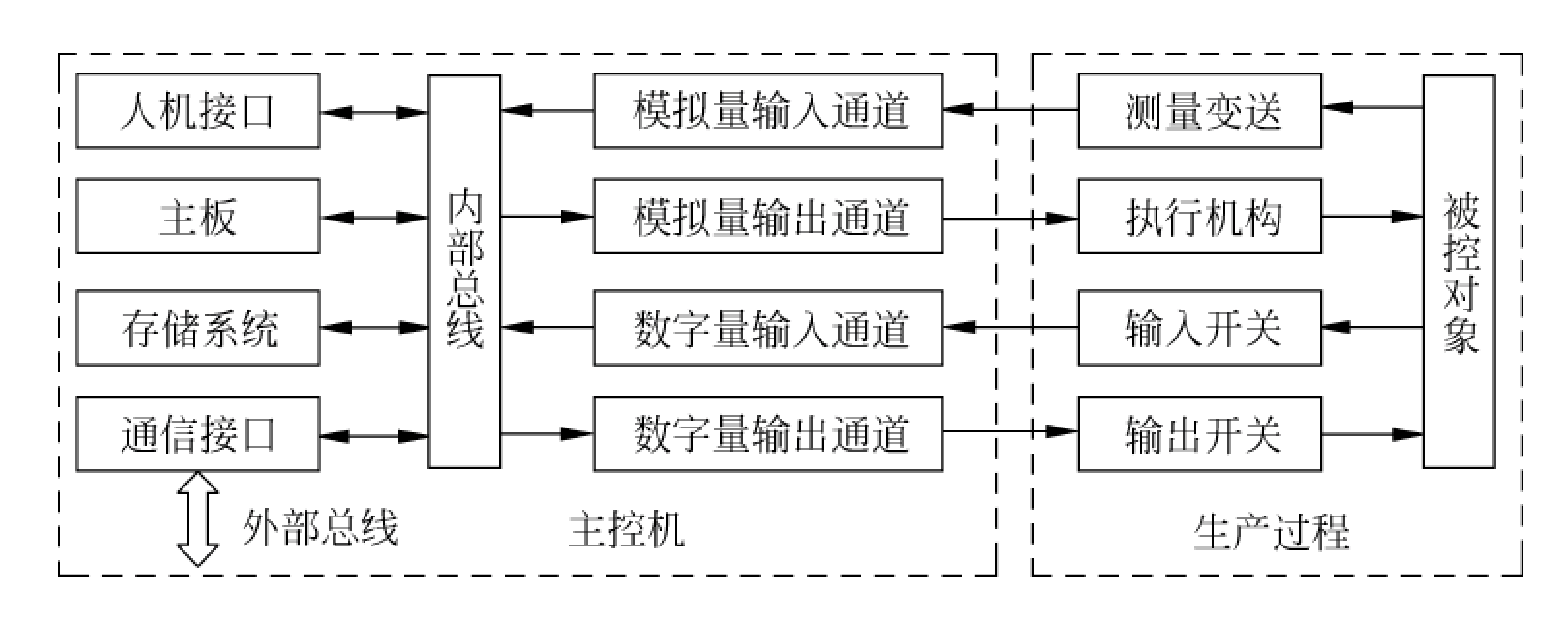


图 3.1 计算机控制系统基本硬件结构

鉴于执行结构和测量变送环节的原理及选型在检测技术及仪表相关课程中已有所介绍,因此,本章主要针对其他几部分内容进行探讨。

3.1 计算机控制系统常用主控制器

在计算机控制系统中,常用主控制器包括:工业控制计算机(IPC)、可编程序控制器(PLC)、智能工业调节器、数字信号处理器(DSP)、单片机(MCU)、ARM 处理器等。

3.1.1　工业控制计算机

工业控制计算机即工控机(Industry Personal Computer,IPC),就是专门为工业现场而设计的计算机,是一种加固的增强型个人计算机,它可以作为一个工业控制器在工业环境中可靠运行,如图 3.2 所示,也叫产业电脑或工业电脑。早在 20 世纪 80 年代初期,美国 AD 公司就推出了类似 IPC 的 MAC-150 工控机,随后美国 IBM 公司正式推出工业个人计算机 IBM7532。由于 IPC 的性能可靠、软件丰富、价格低廉,在控制系统中应用日趋广泛。已被广泛应用于钢铁冶金、石油化工、机电成套设备、医药食品、数控机床、工业炉窑等工业领域,以及军工和科研设备中,取代了 PC 在工业上不足的领域。



(a) 台式



(b) 壁挂式



(c) 便携式

图 3.2　工控机

工控机在架构上与 PC 类似,在用户使用端和 PC 相同,但与 PC 产业的发展却完全是不同的道路。PC 的设计理念是追求更高的速度、更好的使用舒适度、更好的用户体验,常常是以时尚的外形、较高的显卡性能、多媒体显示性能、丰富的扩展性能、多声道声卡等方面作为吸引消费者的卖点,而工控机是专门为工业控制所设计,主要以满足实现工业生产过程的自动控制为目的,注重生产设备的软硬件接口,强调设备之间如何自动沟通,也讲求在工作场所的实际操作,符合在真正作业环境中的各种相关要求。更重要的是它具有丰富功能的组合,可适应广泛的应用层面。例如一个标准的工控机配置如下。

- 机箱:尺寸($W \times H \times D$)为 482mm×177mm×452 mm(19"×7"×17.8")。
- 底板:尺寸 323mm×300mm。
- 槽数:8ISA/4PCI/2CPU 槽。
- 主板:尺寸 338mm×122mm,工控主板是高性能的 Intel Pentium 4 CPU 工控卡,系统核心采用 Intel 845GL/GV DDR 芯片,集成了 2D/3D VGA 功能,高速 USB 2.0,Winbond PCI - ISA brige,及可选双 Intel 82559 以太网控制器和小型 P4 冷却风扇。
- 电源:尺寸 15mm×14mm×8.5mm,工业电源 250W,有 AT、ATX 电源。
- CPU:标准使用 Pentium 4 2.0、Pentium 4 2.4、Pentium 4 2.6、Pentium 4 2.8、Pentium 4 3.0。
- 内存:DDR400,标准使用 512MB～1GB 内存。
- 硬盘:标准使用 IDE 接口,7200/分钟。
- 光驱:普通均可。
- 软驱:普通均可。

- 键鼠：普通均可。
- 网卡：10/100Mbps，可支持 1000Mbps。
- 显卡：标准使用 64MB(主板集成)。
- 标准的工控机外接口有：两个串口，一个并口，两个 USB 口，一个 VGA 接口，一个网口。
- 环境要求：工作温度 0～50℃，存放温度－20～80℃，相对湿度 10%～90%(无自凝)。

另外，工业现场一般具有强烈的震动，灰尘多，另有很高的电磁干扰，而且一般连续作业很少停机休息。因此，工控机与 PC 相比需具有以下特点。

(1) 机箱采用钢材结构，有较高的防磁、防尘、防冲击、防电击的能力。

(2) 可以在差距很大的温度和湿度条件下操作。

(3) 备有系统重置开关，符合人体工学需求。

(4) 机箱内有专用底板，底板上有 PCI 和 ISA 插槽，便于插接扩展板卡。

(5) 机箱内有专门电源，电源有较强的抗干扰能力。

(6) 有较好的散热能力，可以连续长时间工作。

(7) 采用便于安装的标准机箱(4U 高度标准机箱较为常见)，有台式、壁挂式、便携式等设计，如图 3.2 所示。

(8) 一般配置硬盘容量小。

(9) 不具备较高的显卡性能、多媒体显示性能等。

同时，为了便于实现工控机与控制过程进行各种形式的信息交换，从而实现测量数据的输入和控制信号的输出，工控机具备较为完善的过程通道，如模拟量输入/输出通道、数字量(开关量)输入/输出通道等。并且其具有极佳的扩展性。工业现场使用工控机做控制核心，会接上多种扩展卡，扩展卡以早期的 ISA 卡、主流 PCI 卡为主，举例来说，某些传统的数控机床厂家，为保障其早期在 ISA 技术上的投资，会采用传统的 ISA 控制卡为主。而一些从事视频监控的厂家，由于 ISA 带宽根本无法满足苛刻的视频带宽需要，必然要采用 PCI 视频捕捉卡，但是也许会同时选择几个 ISA 界面的 IO 卡配合摄像头云台使用。因此用户对 ISA 和 PCI 插槽的需求多种多样。

3.1.2 可编程序控制器

可编程序控制器是指可通过编程或软件配置改变控制对策的控制器。可编程序控制器(Programmable Controller，PC)经历了可编程序矩阵控制器(Programmable Matrix Controller，PMC)、可编程序顺序控制器(Programmable Sequence Controller，PSC)和可编程序逻辑控制器(Programmable Logic Controller，PLC)几个不同时期。为与个人计算机(PC)相区别，现在常用 PLC 这个名字。

根据 1987 年国际电工委员会(International Electrical Committee)颁布的 PLC 标准草案，PLC 是一种专门为在工业环境下应用而设计的数字运算操作的电子装置。它采用可以编制程序的存储器，用来在其内部存储执行逻辑运算、顺序运算、计时、计数和算术运算等操作的指令，并能通过数字式或模拟式的输入和输出，控制各种类型的机械或生产过程，如图 3.3 所示。

从结构上分，PLC 分为整体式和组合式(模块式)两种。整体式 PLC 包括 CPU 板、I/O

板、显示面板、内存块、电源等，这些元素组合成一个不可拆卸的整体，结构非常紧凑。它的体积小、价格低，小型 PLC 一般采用整体结构。模块式 PLC 包括 CPU 模块、I/O 模块、内存、电源模块、底板或机架，这些模块可以按照一定规则组合配置。

PLC 具有可靠性高，抗干扰能力强，硬件配套齐全，功能完善，易学易用等特点；同时，系统的设计、安装、调试工作量小，维护方便，容易改造；另外体积小，重量轻，能耗低。目前 PLC 在国内外已广泛应用于钢铁、石油、化工、电力、建材、机械制造、汽车、轻纺、交通运输、环保及文化娱乐等各个行业。主要包括以下几方面。



(a) 整体式



(b) 组合式

图 3.3 可编程序控制器

1. 开关量的控制

PLC 取代传统的继电器电路，实现逻辑控制、顺序控制，既可用于单台设备的控制，也可用于多机群控及自动化流水线。如注塑机、印刷机、订书机械、组合机床、磨床、包装生产线、电镀流水线等。

2. 模拟量控制

温度、压力、流量、液位和速度等都是模拟量，为了使可编程序控制器处理模拟量，必须实现模拟量(A)和数字量(D)之间的 A/D 转换及 D/A 转换。PLC 厂家都生产配套的 A/D 和 D/A 转换模块，使可编程控制器用于模拟量控制。

3. 运动控制

PLC 可以用于圆周运动或直线运动的控制。从控制机构配置来说，早期直接用于开关量 I/O 模块连接执行机构和位置传感器，现在一般使用专用的运动控制模块。如可驱动步进电机或伺服电机的单轴或多轴位置控制模块等。

4. 过程控制

过程控制是指对温度、压力、流量等模拟量的闭环控制。PLC 能编制各种各样的控制算法程序，完成闭环控制。例如 PID 调节是一般闭环控制系统中用得较多的调节方法。大中型 PLC 都有 PID 模块，目前许多小型 PLC 也具有此功能模块。过程控制在冶金、化工、热处理、锅炉控制等场合有非常广泛的应用。

5. 数据处理

现代 PLC 具有数学运算(含矩阵运算、函数运算、逻辑运算)、数据传送、数据转换、排

序、查表、位操作等功能,可以完成数据的采集、分析及处理。这些数据可以与存储在存储器中的参考值比较,完成一定的控制操作,也可以利用通信功能传送到别的智能装置,或将它们打印制表。数据处理一般用于大型控制系统,如无人控制的柔性制造系统;也可用于过程控制系统,如造纸、冶金、食品工业中的一些大型控制系统。

6. 通信及联网

PLC 通信含 PLC 间的通信及 PLC 与其他智能设备间的通信。随着计算机控制的发展,工厂自动化网络发展得很快,各 PLC 厂商都十分重视 PLC 的通信功能,纷纷推出各自的网络系统。新近生产的 PLC 都具有通信接口,通信简单、方便。

3.1.3 智能工业调节器

智能工业调节器是一种基于数字集成技术的过程控制智能型仪表,采用专用微处理器或单片机作为调节器的核心,硬件采用电擦除存储器、开关电源、高速 A/D 转换器等设计而成,常嵌入有成熟的 PID 算法及 PID 自整定技术,从而保证智能型工业调节器具有测控精度高,抗干扰性能强,功能完备,操作简单等特点,具有广泛的适用性,外形如图 3.4 所示。



图 3.4 智能调节器

硬件多采用模块方式设计,可根据需要进行灵活配置;软件采用组态方式,可通过按键操作对仪表功能进行组态编程,以实现多种信号类型输入和多种调节、控制输出。因此其适用范围非常广泛,不但可应用于各种温度测控系统,而且还可应用于湿度、压力、称重、流量、液位、酸度等多种工业测控领域。既可以作为控制器配合多种执行器,实现对电加热设备和电磁、电动、气动阀门等进行 PID 控制。也可以作为变送器实现对温度、压力、流量、液位等物理量的测量、显示、报警控制和变送输出。具有完善的网络通信功能,带有多种通信模块,如 RS-485、RS-232、RS-422 数字通信等,可与计算机进行高速、高效的双向数据交换。具备多重保护、隔离设计、抗干扰能力强、可靠性高。因此在工业集散控制系统中作为工作站,是一种非常理想的产品。另外,还可实现自动/手动控制方式的双向无扰切换等。

3.1.4 数字信号处理器

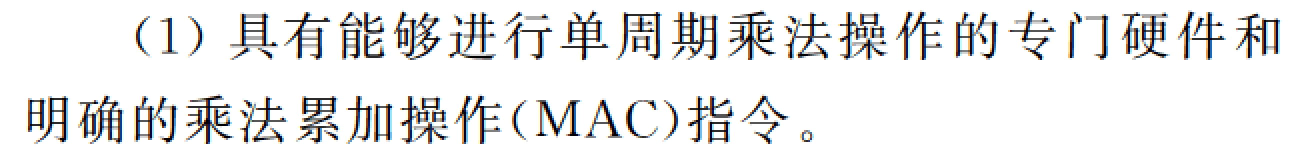
数字信号处理器(Digital Signal Processor,DSP)是进行数字信号实时处理的微处理

器，是伴随着微电子学、数字信号处理技术、计算机技术的发展而产生的器件，如图 3.5 所示。它可以快速地实现对信号的采集、变换、滤波、估值、增强、压缩、识别等处理，以得到符合人们需要的信号形式。DSP 处理器与诸如英特尔奔腾或 Power PC 的通用处理器有很大的区别，主要是由于 DSP 的结构和指令是专门针对信号处理而设计和开发的，它具有以下特点。



图 3.5　数字信号处理器

(1) 具有能够进行单周期乘法操作的专门硬件和明确的乘法累加操作(MAC)指令。



(2) 采用哈佛结构，有两个存储空间，即程序存储空间和数据存储空间。处理器内核通过两套总线与这些存储空间相连，允许对两存储器同时进行访问，使处理器的带宽加倍。

(3) 具有零消耗循环控制的专门硬件。

(4) DSP 通常包含有专门的地址产生器，它能产生信号处理算法需要的特殊寻址，如循环寻址和位翻转寻址。循环寻址对应于流水 FIR(Finite Impulse Response)滤波算法，位翻转寻址对应于 FFT(Fast Fourier Transform)算法。

(5) DSP 执行程序的进程对程序员来说是透明的，即执行时间具有可预测性，因此很容易预测处理每项工作的执行时间。

(6) DSP 处理器具有 DMA、串口、Link 口、定时器等丰富的外设。

DSP 处理器的品种繁多，四大 DSP 生产商主要有：Texas Instruments(德州仪器)公司、Lucent Technologies(朗讯技术)公司、Analog Devices(模拟设备)公司和 Motorola(摩托罗拉)公司，其中 TI 占有最大的市场份额。

DSP 处理器可分为两大类：定点 DSP 和浮点 DSP。定点 DSP 发展迅速，品种最多，处理速度为 20～2000MIPS。浮点 DSP 基本由 TI 和 AD 公司垄断，处理速度 40～1000MFLOPS。DSP 的性能已形成低、中、高三档，高端产品处理器结构发生了深刻的变化，形成了多样化的趋势。

数字信号处理器主要应用于通信与信息系统、信号与信息处理、自动控制、雷达、军事、航空航天、医疗、家用电器等许多领域。

语音处理：语音编码、语音合成、语音识别、语音增强、语音邮件、语音储存等。

图像/图形：二维和三维图形处理、图像压缩与传输、图像识别、动画、机器人视觉、多媒体、电子地图、图像增强等。

军事：保密通信、雷达处理、声呐处理、导航、全球定位、跳频电台、搜索和反搜索等。

仪器仪表：频谱分析、函数发生、数据采集、地震处理等。

自动控制：控制、自动驾驶、机器人控制、磁盘控制等。

医疗：助听、超声设备、诊断工具、病人监护、心电图等。

家用电器：数字音响、数字电视、可视电话、音乐合成、音调控制、玩具与游戏等。

在工程应用中，DSP 的选型主要考虑处理速度、功耗、程序存储器和数据存储器的容量与片内的资源，如定时器的数量、I/O 口数量、中断数量、DMA 通道数等。以 TI 公司 DSP 为例，目前主要有四大系列的 DSP，其主要特点比较如下。

1. C5000 系列(定点、低功耗)

C54X,C54XX,C55XX 相比其他系列的主要特点是低功耗,所以最适合个人与便携式上网以及无线通信应用,如手机、PDA、GPS 等应用。处理速度在 80～400MIPS 之间。C54XX 和 C55XX 一般只具有 McBSP 同步串口、HPI 并行接口、定时器、DMA 等外设。值得注意的是 C55XX 提供了 EMIF 外部存储器扩展接口,可以直接使用 SDRAM,而 C54XX 则不能直接使用。两个系列的数字 IO 都只有两条。

2. C2000 系列(定点、控制器)

C20X,F20X,F24X,F24XX,C28X 系列芯片具有大量外设资源,如:A/D、定时器、各种串口(同步和异步),WATCHDOG、CAN 总线、PWM 发生器、数字 IO 等。是针对控制应用最佳化的 DSP,在 TI 所有的 DSP 中,只有 C2000 有 Flash,也只有该系列有异步串口可以和 PC 的 UART 相连。

3. C6000 系列

C62XX,C67XX,C64X 系列以高性能著称,最适合宽带网络和数字影像应用。其中:C62XX 和 C64X 是定点系列,C67XX 是浮点系列。该系列提供 EMIF 扩展存储器接口。该系列只提供 BGA 封装,只能制作多层 PCB,且功耗较大。

4. OMAP 系列

OMAP 处理器集成 ARM 的命令及控制功能,另外还提供 DSP 的低功耗实时信号处理能力,最适合移动上网设备和多媒体家电。

在选型使用时,一定要根据项目需要选择使用。

3.1.5 单片机

单片机是典型的嵌入式微控制器(MicroController Unit,MCU),是采用超大规模集成电路技术把具有数据处理能力的中央处理器 CPU、随机存储器 RAM、只读存储器 ROM、多种 I/O 口、中断系统、定时器/计时器等功能集成到一块硅片上构成的一个小而完善的计算机系统。有些型号的单片机甚至还包括显示驱动电路、脉宽调制电路、模拟多路转换器、A/D 转换器等功能电路等。如图 3.6 所示为一款 PIC 单片机。

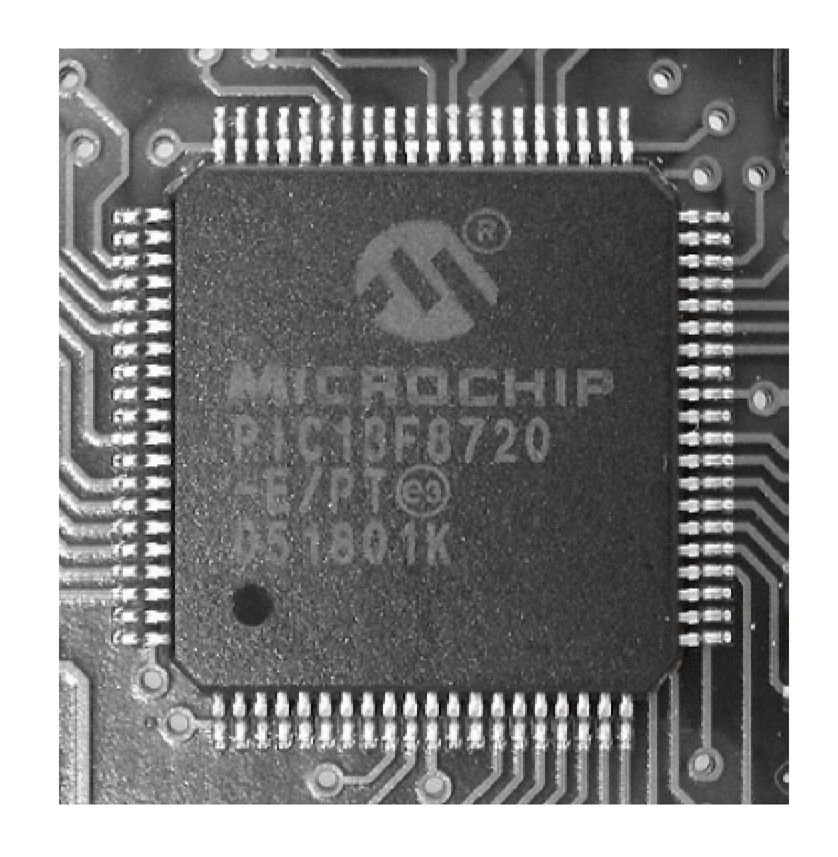


图 3.6　PIC 单片机

目前单片机的应用已渗透到我们生活的各个领域,如自动控制领域的机器人、智能仪表、医疗器械、导弹的导航装置、飞机上各种仪表、计算机的网络通信与数据传输设备、工业自动化的实时控制和数据处理设备、各种智能 IC 卡、民用豪华轿车的安全保障系统、录像机、摄像机、全自动洗衣机的控制电路,以及程控玩具、电子宠物等。

单片机的种类非常多,常用单片机有几下几种。

(1) STC 单片机：STC 公司的单片机主要是基于 8051 内核，是新一代增强型单片机，指令代码完全兼容传统 8051，速度快 8～12 倍，带 ADC、4 路 PWM、双串口，有全球唯一 ID 号，加密性好，抗干扰性强。

(2) PIC 单片机：是 MICROCHIP 公司的产品，其突出的特点是体积小、功耗低、精简指令集，抗干扰性好，可靠性高，有较强的模拟接口，代码保密性好，大部分芯片有其兼容的 Flash 程序存储器的芯片。

(3) EMC 单片机：是台湾义隆公司的产品，有很大一部分与 PIC 8 位单片机兼容，且相兼容产品的资源相对比 PIC 多，价格便宜，有很多系列可选，但抗干扰性较差。

(4) ATMEL 单片机(51 单片机)：ATMEL 公司的 8 位单片机有 AT89、AT90 两个系列，AT89 系列是 8 位 Flash 单片机，与 8051 系列单片机相兼容，静态时钟模式；AT90 系列单片机是增强 RISC 结构，全静态工作方式，内载在线可编程 Flash 的单片机，也叫 AVR 单片机。

(5) PHILIPS 51LPC 系列单片机(51 单片机)：PHILIPS 公司的单片机是基于 80C51 内核的单片机，嵌入了掉电检测、模拟以及片内 RC 振荡器等功能，这使 51LPC 在高集成度、低成本、低功耗的应用设计中可以满足多方面的性能要求。

(6) TI 公司单片机(51 单片机)：德州仪器提供了 TMS370 和 MSP430 两大系列通用单片机。TMS370 系列单片机是 8 位 CMOS 单片机，具有多种存储模式和多种外围接口模式，适用于复杂的实时控制场合；MSP430 系列单片机是一种超低功耗、功能集成度较高的 16 位低功耗单片机，特别适用于要求功耗低的场合。

3.1.6 ARM 处理器

ARM(Advanced RISC Machines)处理器由 ARM 公司设计，采用发放许可方式，由其他公司生产。它是一个 32 位精简指令集 RISC(Reduced Instruction Set Computer，精简指令集计算机)处理器架构，具有性能高、成本低和耗能省的特点，适用于多种领域，如嵌入式实时控制系统、消费或教育类多媒体、移动式应用设备等。ARM 内核分为 ARM7、ARM9、ARM10 以及 StrongARM 等几类。每一类又根据各自包含的功能模块而具有多种构成。

ARM7 是 32 位低价位、低功耗内核，平均功耗为 0.6mW/MHz，具有 0.9MIPS/MHz 的三级流水线，冯·诺依曼结构，时钟速度为 66MHz，每条指令平均执行 1.9 个时钟周期。是 ARM 公司最早为业界普遍认可且得到了最为广泛应用的处理器核，而已是目前最低端的 ARM 核。

ARM9 内核具有高性能和低功耗的特点，平均功耗为 0.7mW/MHz，五级流水线，提供 1.1MIPS/MHz 的哈佛结构，时钟速度为 120～200MHz，每条指令平均执行 1.5 个时钟周期。

ARM10 采用 64 位 AHB(Advanced High Performance Bus，高性能总线)数据接口，1.25MIPS/MHz 的哈佛结构，六级流水处理，平均功耗为 1000mW，时钟速度为 300MHz，每条指令平均执行 1.2 个周期。

ARM10TDMI 与所有 ARM 核在二进制级代码兼容，内带高速 32×16MAC，预留 DSP 协处理器接口。其中的 VFP10(矢量浮点单元)为七级流水结构。

ARM1020T 处理器是由 ARM10TDMI、32KB 指令和数据 Caches(高速缓冲存储器)及

MMU(Memory Management Unit,内存管理单元)几部分构成,64 位,300MHz 时钟,功耗为 1W(2.0V 供电),能够支持多种商用操作系统,适用于高性能手持式因特网设备及数字式消费类产品。

ARM11 系列微处理器是 ARM 公司近年推出的新一代 RISC 处理器,是 ARM 家族中性能最强的一个系列,它是 ARM 新指令架构 ARMv6 的第一代设计实现。该系列主要有 ARM1136J、ARM1156T2 和 ARM1176JZ 三个内核型号,分别针对不同应用领域。

各类 ARM 处理器流水线如图 3.7 所示。

ARM7	预取 (Fetch)	译码 (Decode)	执行 (Exec)					
ARM9	预取 (Fetch)	译码 (Decode)	执行 (Exec)	访问 (Memory)	写回 (Write)			
ARM10	预取 (Fetch)	发射 (Lssue)	译码 (Decode)	执行 (Exec)	访问 (Memory)	写回 (Write)		
ARM11	预取 (Fetch)	预取 (Fetch)	发射 (Lssue)	译码 (Decode)	转换 (Snny)	执行 (Exec)	访问 (Memory)	写回 (Write)

图 3.7 ARM 处理器流水线

3.2 总线技术

所谓总线就是计算机各功能部件之间互连和传送信息的一组信号线,按照所传输的信息种类,总线可以划分为数据总线、地址总线和控制总线,分别用来传输数据、数据地址和控制信号。

总线按功能和规范可分为三大类型:片总线、内总线和外总线。

(1) 片总线(Chip Bus, C-Bus)又称元件级总线,是把各种不同的芯片连接在一起构成特定功能模块(如 CPU 模块)的信息传输通路。包括数据总线、地址总线、控制总线、I^2C(Inter-Integrated Circuit)总线、SPI 总线(Serial Peripheral Interface,串行外设接口)、SCI 总线(Serial Communication Interface,串行通信接口)等。

(2) 内总线(Internal Bus, I-Bus)又称系统总线或板级总线,是微机系统中各插件之间的信息传输通路,即微机主板插槽(系统扩展板)遵循的各种标准。例如 CPU 模块和存储器模块或 I/O 接口模块之间的传输通路。包括 ISA (Industry Standard Architecture)总线、EISA (Extended Industry Standard Architecture) 总线、VESA (Video Electronic Standard Association)总线、PCI 总线(Peripheral Component Interconnect)等。

(3) 外总线(External Bus, E-Bus)又称通信总线,是微机系统之间或微机系统与其他系统(仪器、仪表、控制装置等)之间信息传输的通路,它用于设备一级的互连,多表现为微机对外的标准接口插头,有时也称为接口标准。包括 RS-232C、RS-485、IEEE-488、USB 等总线。另外,在工业控制中,还定义了其他总线,如 VME(Versa Module Eurocard)、STD、PC-104、Compact PCI 等。

以上三类总线在计算机控制系统中的关系如图 3.8 所示。

另外,按照传输数据的方式,总线又可划分为串行总线和并行总线。

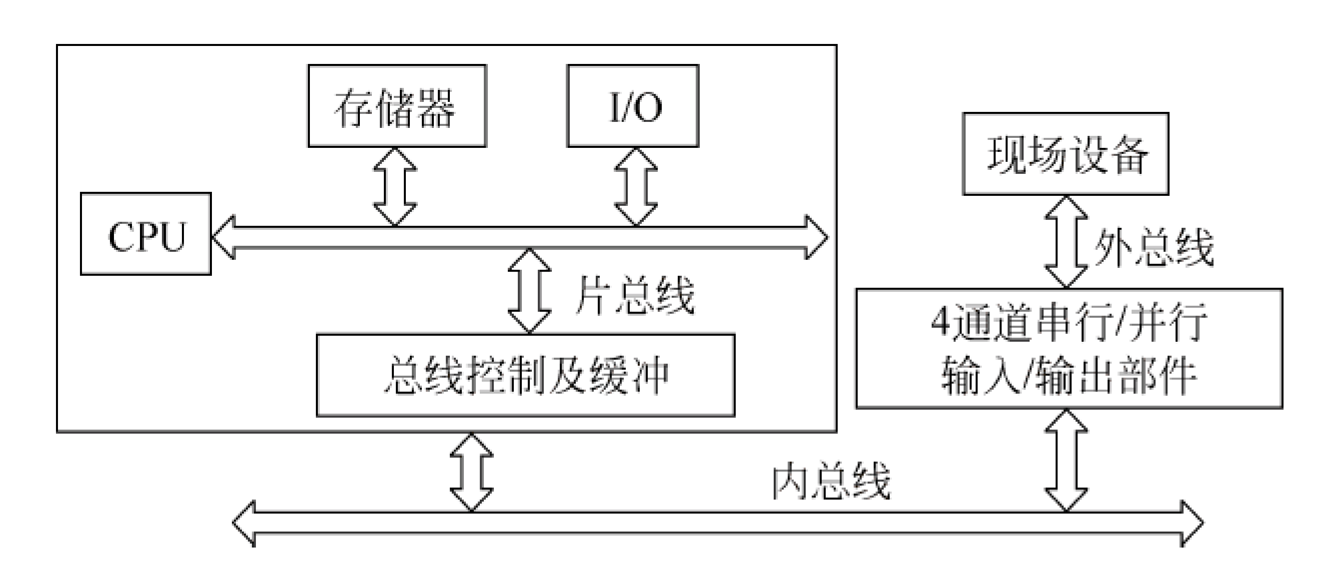


图 3.8　三类总线在计算机控制系统中的关系

串行总线用于串行通信,外设和计算机间使用一根数据信号线(另外需要地线,可能还需要控制线),数据在一根数据信号线上一位一位地进行传输,每一位数据都占据一个固定的时间长度。这种通信方式使用的数据线少,在远距离通信中可以节约通信成本,当然,其传输速度比并行传输慢。常见的串行总线有 I^2C 总线、SPI 总线、USB、RS-232、RS-422、RS-485 等。

并行总线就是并行接口与计算机设备之间传递数据的通道,采用并行传送方式在计算机与外部设备之间进行数据传送的接口叫并行接口,它有两个主要特点:一是同时并行传送的二进制位数就是数据宽度;二是在计算机与外设之间采用应答式的联络信号来协调双方的数据传送操作,这种联络信号又称为握手信号。这种通信方式传输速度快,常用于近距离通信。常见的并行总线有 ISA 总线、PCI 总线等。

评价一种总线的性能主要注意以下几个方面参数。

(1) 总线时钟频率:总线的工作频率,以 MHz 表示,它是影响总线传输速率的重要因素之一。

(2) 总线宽度:数据总线的位数,用位(b)表示,如总线宽度为 8 位、16 位、32 位和 64 位。

(3) 总线传输速率:在总线上每秒钟传输的最大字节数(MB/s),即每秒处理多少兆字节。

可以通过总线宽度和总线时钟频率来计算总线传输速率(带宽):传输速率=总线时钟频率×总线宽度/8,如某主板 PCI 总线宽度 16 位,总线频率 66MHz,总线数据传输速率=66×16/8(MB/s)=132(MB/s)。

下面针对计算机控制技术中常用的几种总线形式做一详细介绍。

3.2.1　片总线

1. I^2C 总线

I^2C 总线是一种由 Philips(飞利浦)公司开发的两线式串行总线,用于连接微控制器及其外围设备。是微电子通信控制领域广泛采用的一种总线标准。它是同步通信的一种特殊形式,具有接口线少,控制方式简单,器件封装形式小,通信速率较高等优点。它只有两条总线线路:一条串行数据线 SDA,一条串行时钟线 SCL。每个连接到总线的器件都可以通过唯一的地址和一直存在的简单的主机/从机关系软件设定地址,主机可以作为主机发送器或

主机接收器。它是一个真正的多主机总线,如果两个或更多主机同时初始化,数据传输可以通过冲突检测和仲裁防止数据被破坏。串行的 8 位双向数据传输位速率在标准模式下可达 100kb/s,快速模式下可达 400kb/s,高速模式下可达 3.4Mb/s;连接到相同总线的 IC 数量只受到总线的最大电容 400pF 限制。

I^2C 总线在传送数据过程中共有三种类型信号,它们分别是:开始信号、结束信号和应答信号。

- 开始信号:SCL 为高电平时,SDA 由高电平向低电平跳变,开始传送数据。
- 结束信号:SCL 为低电平时,SDA 由低电平向高电平跳变,结束传送数据。
- 应答信号:接收数据的 IC 在接收到 8b 数据后,向发送数据的 IC 发出特定的低电平脉冲,表示已收到数据。CPU 向受控单元发出一个信号后,等待受控单元发出一个应答信号,CPU 接收到应答信号后,根据实际情况做出是否继续传递信号的判断。若未收到应答信号,则判断为受控单元出现故障。

2. SPI 总线

SPI 高速同步串行口是一种标准的四线同步双向串行总线。SPI 接口主要应用在 EEPROM、Flash、实时时钟、A/D 转换器,还有数字信号处理器和数字信号解码器之间。在芯片的管脚上只占用 4 根线,节约了芯片的管脚,同时为 PCB 的布局节省空间,很多芯片集成了这种通信协议。

SPI 以主从方式工作,这种模式通常有一个主设备和一个或多个从设备,需要 4 根连线;用于单向传输时,也就是半双工方式时,3 根线也可以。这是所有基于 SPI 的设备共有的,4 根信号线如下。

(1) SDO(数据输出):主设备数据输出,从设备数据输入。

(2) SDI(数据输入):主设备数据输入,从设备数据输出。

(3) SCK(时钟):时钟信号,由主设备产生。

(4) CS(片选):从设备使能信号,由主设备控制。

CS 是控制芯片是否被选中,另外 3 根线负责通信:由 SCK 提供时钟脉冲,SDI、SDO 则基于此脉冲完成数据传输。数据输出通过 SDO 线,数据在时钟上升沿或下降沿时改变,在紧接着的下降沿或上升沿被读取。完成一位数据传输,输入也使用同样原理。经过至少 8 次时钟信号的改变(上升沿和下降沿为一次),就可以完成 8 位数据的传输。要注意的是:SCK 信号线只由主设备控制。

3.2.2 内总线

1. ISA/EISA 总线

1) ISA/EISA 总线概述

ISA 总线又称 AT 总线,是 IBM 公司为 PC/AT 电脑而制定的总线标准,为 16 位体系结构,支持 8/16 位的 I/O 设备,最大传输率为 16MB/s,总线频率 8MHz,也称为 AT 标准。1988 年康柏、HP、NEC 等 9 个厂商协同把 ISA 扩展到 32 位,即 EISA 总线,总线频率达到 16MHz,与 ISA 在结构上具有良好的兼容性。虽然以上两种总线形式已逐渐被

PCI 总线所替代，但在工业控制中仍有一些设备中还在应用。下面对 ISA 总线做一简单介绍。

ISA 总线扩展插槽由两部分组成，一部分为 8 位基本插槽，有 62 引脚，正反面分别称为 A 列和 B 列，其信号分布及名称与 PC/XT 总线的扩展槽基本相同；另一部分是 AT 机的添加部分，由 36 引脚组成，正反面分别称为 C 列和 D 列，它增加了 8 位数据线和 7 位地址线，不能独立工作，需和基本插槽一起使用。具有 ISA 总线的 PCB 板卡及 ISA 插槽如图 3.9 所示。ISA 插槽引脚如图 3.10 所示。

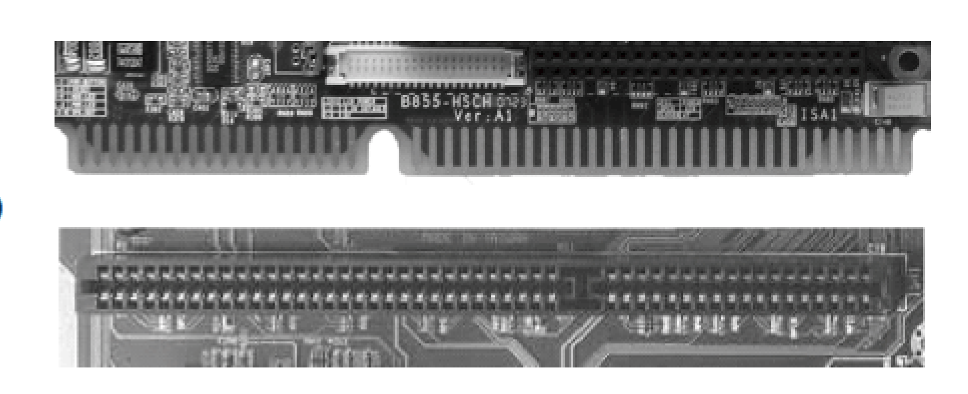


图 3.9　具有 ISA 总线的 PCB 板卡及 ISA 插槽

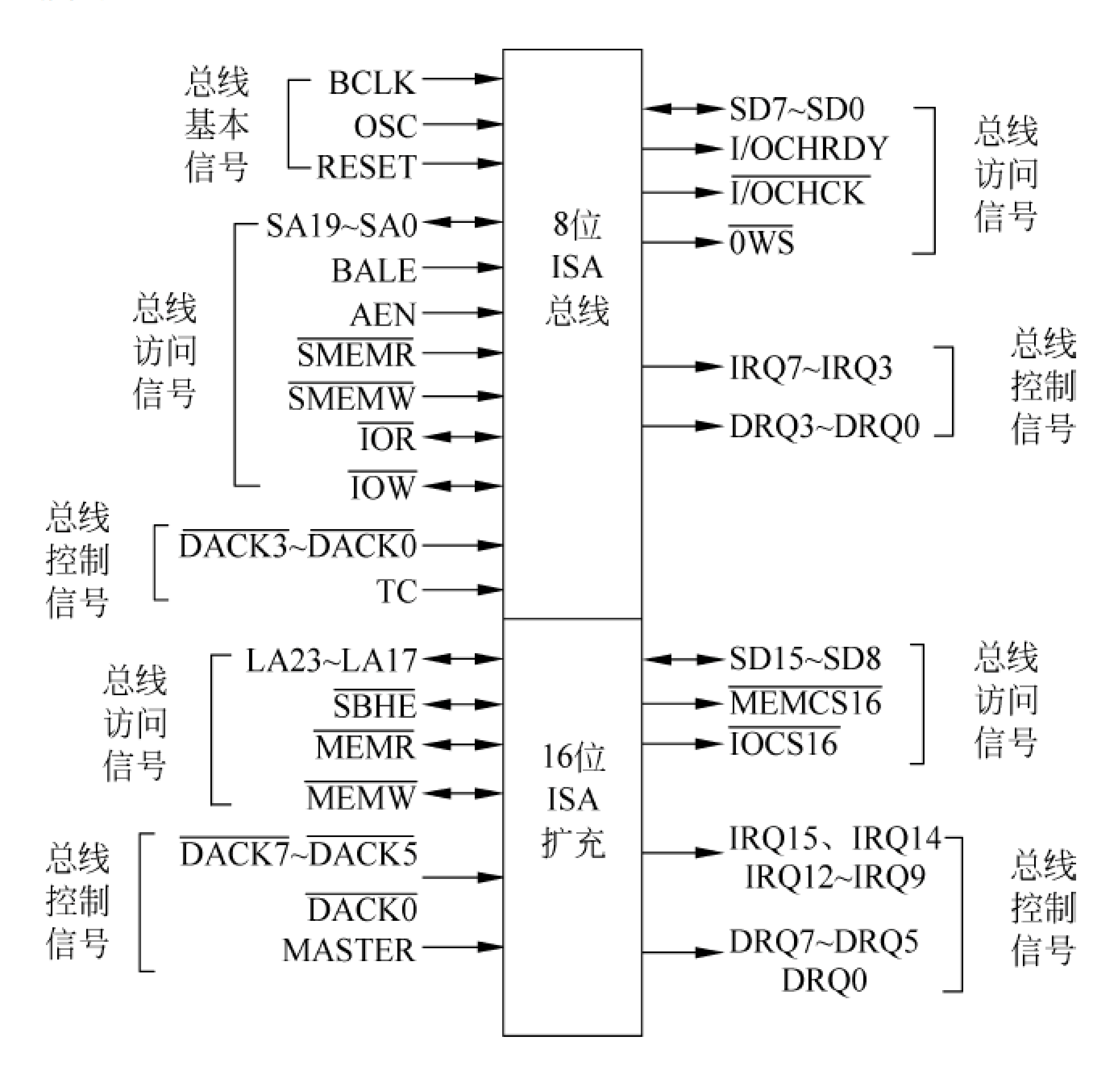


图 3.10　ISA 插槽引脚

ISA 总线基本引脚定义如下。

- RESET、BCLK：复位及总线基本时钟，BCLK＝8MHz。
- OSC：一个 70 毫微秒的时钟(14.31818 MHz)，该信号与系统时钟不同步。
- SA19～SA0：存储器及 I/O 空间 20 位地址，带锁存。
- BALE：用来锁存 LA23～LA17 信号或者译码这些信号，BALE 下降沿用于锁存 LA23～LA17，在 DMA 周期中 BALE 被强制为高，此信号与 AEN 并用时表明一个有效的微处理器或 DMA 地址。
- AEN：地址允许，高电平有效时表明 CPU 让出总线用于 DMA 传送。即当 AEN 有效时总线上的地址、数据和读写信号由 DMA 控制器控制。ISA 扩展板的片选译码应包含 AEN 信号以防止 DMA 周期中出现不正确的片选。
- $\overline{\text{SMEMR}}$、$\overline{\text{SMEMW}}$：8 位 ISA 存储器读写控制。
- $\overline{\text{IOR}}$、$\overline{\text{IOW}}$：ISA 总线的 I/O 读写控制。

- $\overline{\text{DACK7}}$～$\overline{\text{DACK5}}$、$\overline{\text{DACK3}}$～$\overline{\text{DACK0}}$：DMA 请求响应。分别被用来确认 DRQ7～DRQ5 和 DRQ3～DRQ0 的 DMA 请求。
- TC：在 DMA 通道操作中当计数完成时产生的终端计数信号。
- LA23～LA17：是系统中内存地址，它们和 SA19～SA0 可以共同寻址多达 16MB 的内存，当 BALE 为高时这些信号才有效。由于它们是非锁存的，故在整个总线周期中它们并不总是保持有效状态，用 BALE 下降沿锁存这些信号的译码。
- $\overline{\text{SBHE}}$：高字节允许，打开 SD15～SD8 数据通路。
- $\overline{\text{MEMR}}$、$\overline{\text{MEMW}}$：16 位 ISA 存储器读写控制。
- MASTER：ISA 主模块确认信号，ISA 发出此信号，与主机内 DMAC 配合使 ISA 卡成为主模块，全部控制总线。
- SD15～SD8：数据总线，访问 8 位 ISA 卡时高 8 位自动传送到 SD7～SD0。
- I/OCHRDY：I/OCHRDY 通常处于高就绪状态，ISA 板将 I/OCHRDY 拉低（未准备好）以插入等待状态。
- $\overline{\text{IOCHCK}}$：ISA 板卡奇偶校验错误信号，由 ISA 板生成进而引发非屏蔽中断，当它有效时表明发现了不可恢复的错误。
- $\overline{\text{0WS}}$：由地址译码产生，不插入任何额外等待状态，即可完成一个总线周期。
- IRQ15、IRQ14、IRQ12～IRQ9、IRQ7～IRQ3：中断请求。
- DRQ7～DRQ5、DRQ3～DRQ0：ISA 卡 DMA 请求。
- $\overline{\text{MEMCS16}}$、$\overline{\text{IOCS16}}$：ISA 卡发出此信号确认可以进行 16 位数据传送。
- REFRESH：该信号为低时表明正在进行内存刷新操作。

2）ISA 总线应用

下面介绍一个 ISA 总线的工程应用实例。如图 3.11 所示是一个用于大型支撑定位系统的伺服电机控制系统结构图，虚线框以内为伺服控制卡部分。伺服控制卡用于产生差动脉冲，使驱动器工作在脉冲方式，控制 8 路电机的运转，实现 8 根绳索的收、放，从而实现系统悬吊定位。这里，脉冲的频率控制电机的转速，脉冲的个数控制电机的转角。同时控制计算机通过伺服控制卡接收现场送来的异常信号，产生中断，及时对异常情况进行处理。系统主要包括控制计算机的 ISA 总线接口单元、地址单元、总线数据单元、脉冲发生单元、中断控制单元、FPGA（Field Programmable Gata Array，现场可编程门阵列）单元，以及与现场进行信号传递的接口单元等。软件设计完成图形界面、数据通信、控制命令等。

控制卡（见图 3.11 虚线框以内）插在控制计算机的标准 ISA 插槽中，地址信号、读/写信号，以及 8 位数据信号由于需要分别送往 8 片 8254 和一片 FPGA，因此通过两片 74LS245 接入 FPGA 单元，在 FPGA 单元内实现板地址译码和全局地址译码，同时也通过 FPGA 进行数据的读/写以及中断的控制。8254 作为脉冲发生单元，用来产生 8 路脉冲信号。脉冲信号再经 4 片 AM26LS31 进行 TTL 信号差分转换后，对电机进行控制，决定电机运转的位移。

现场送来的 8 路异常信号和暂停信号通过光电隔离和 FPGA 内部的数据接口单元供控制计算机进行查询，同时，有任何异常信号或暂停信号到来时，通过 FPGA 内部的中断控制单元向 CPU 发出中断申请。另外，控制计算机通过 FPGA 内的接口电路对现场电机进行开/关伺服的控制，信号通过光电隔离送往控制现场。

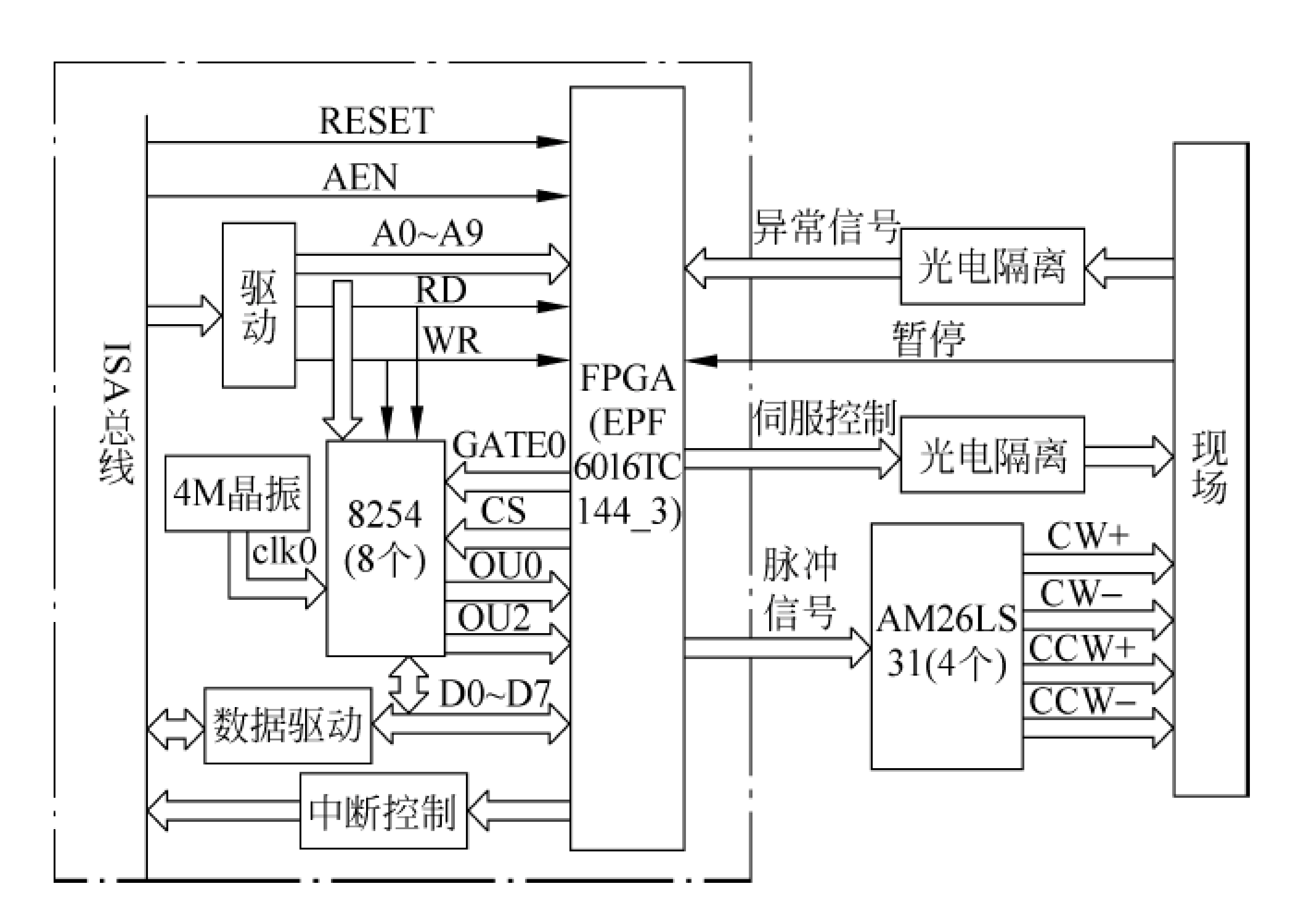


图 3.11 基于 ISA 总线的伺服控制卡硬件结构图

这里，我们选用了松下 A 系列 MDMA102 型和 MHMA502 型交流伺服电动机，以及 MDDA103 型和 MHDA503 型驱动器。

2. PCI/Compact PCI 总线

1）PCI 总线简介

通常认为，I/O 总线的速度应为外设速度的 3～5 倍。因此，随着外设速度的提高，对总线的速度提出了更高的要求，原有的 ISA 和 EISA 已远远不能满足需要。

PCI 总线从 1992 年创立规范至今，已成为了计算机的一种标准总线，取代了早先的 ISA 总线，如图 3.12 所示为 PCI 总线及系统主板上的 PCI 插槽。当然在 PCI 总线后又出现了专门用于显卡的 AGP（Accelerated Graphics Port，加速图形接口）总线，以及 PCI Express 总线，但是 PCI 本身具有许多优点，如即插即用（Plug and Play），即当板卡插入系统时，系统会自动对板卡所需资源进行分配，如基地址、中断号等，并自动寻找相应的驱动程序，而 ISA 板卡却需要进行手动配置。因此 PCI 总线得到了非常广泛的应用。

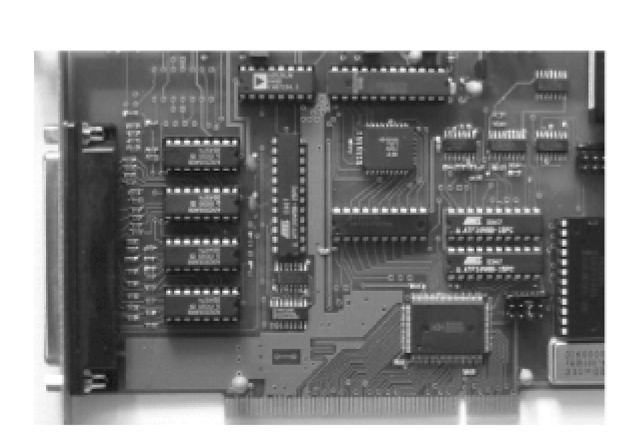
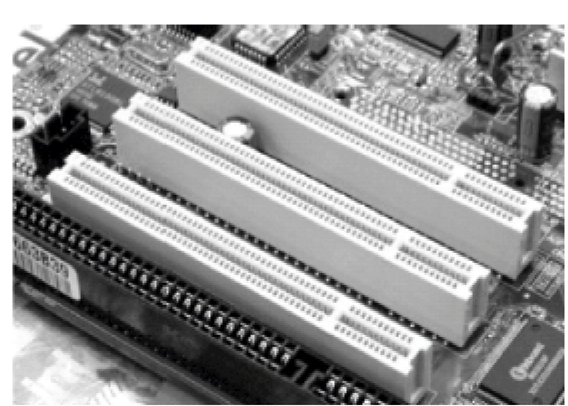



图 3.12 PCI 总线及插槽

从数据宽度上看，PCI 总线有 32 位（5V）、64 位（3.3V）之分；从总线速度上分，有 33MHz、66MHz 两种。改良的 PCI 系统 PCI-X 最高可以达到 133MHz，这样就可以得到超过 1GB/s 的数据传输速率。

PCI 是由 Intel 公司 1991 年推出的一种局部总线。从结构上看，PCI 是在 CPU 和原来的系统总线之间插入的一级总线，具体由一个桥接电路实现对这一层的管理。桥就是一个

总线转换器和控制器，习惯上将与 CPU 连接的芯片称为北桥，与 I/O 设备连接的芯片称为南桥，并实现上下之间的接口以协调数据的传送。管理器提供了信号缓冲，使之能支持 10 种外设，并能在高时钟频率下保持高性能，它为显卡、声卡、网卡、Modem 等设备提供了连接接口。基于 PCI 总线的系统基本构架如图 3.13 所示。

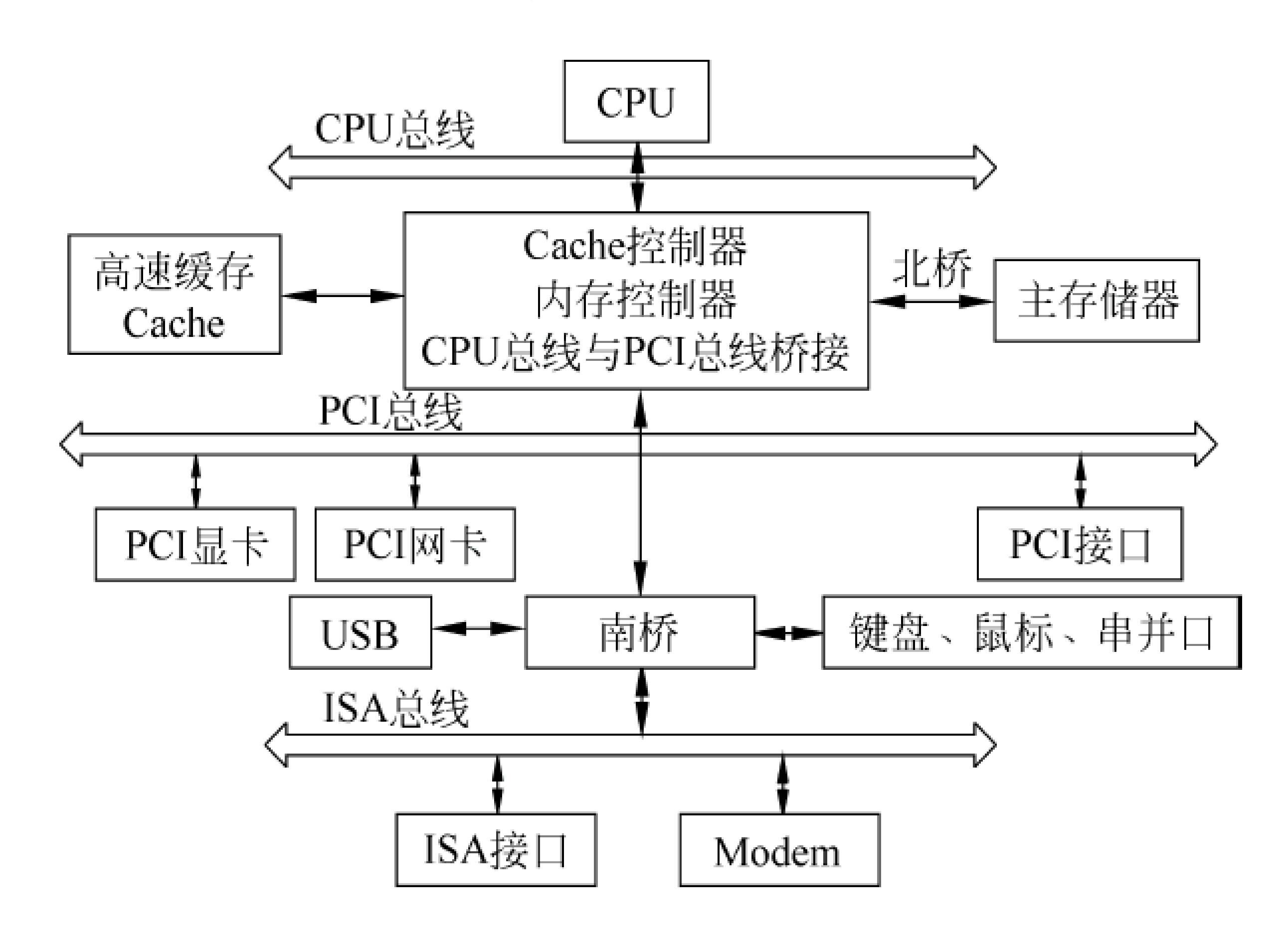


图 3.13 基于 PCI 总线的系统基本构架

PCI 总线也支持总线主控技术，允许智能设备在需要时取得总线控制权，以加速数据传送。总线在数据传输时，由一个 PCI 设备做发起者(主控，Master)，而另一个 PCI 设备做目标(从设备，Slave)。总线上的所有时序的产生与控制，都由 Master 来发起。PCI 总线在同一时刻只能供一对设备完成传输，这就要求有一个仲裁机构(Arbiter)来决定谁掌握总线的主控权。如当 PCI 总线进行操作时，发起者(Master)先置 $\overline{\text{REQ}}$，当得到仲裁器(Arbiter)的许可($\overline{\text{GNT}}$)时，会将 $\overline{\text{FRAME}}$ 置低，并在 AD 总线上放置 Slave 地址，同时 $\overline{\text{C/BE}}$ 放置命令信号，说明接下来的传输类型。所有 PCI 总线上设备都需对此地址译码，被选中的设备要置 $\overline{\text{DEVSEL}}$ 以声明自己被选中。然后当 $\overline{\text{IRDY}}$ 与 $\overline{\text{TRDY}}$ 都置低电平时，可以传输数据。当 Master 数据传输结束前，将 $\overline{\text{FRAME}}$ 置高以表明只剩最后一组数据要传输，并在传完数据后放开 $\overline{\text{IRDY}}$ 以释放总线控制权。

这里我们可以看出，PCI 总线的传输是很高效的，发出一组地址后，理想状态下可以连续发数据，峰值速率为 132MB/s。实际上，目前流行的 33MHz、32b 北桥芯片一般可以做到 100MB/s 的连续传输。

另外，不同于 ISA 总线，PCI 总线的地址总线与数据总线是分时复用的。这样一方面可以节省接插件的管脚数，另一方面便于实现突发数据传输。如图 3.14 所示为一方面与 PCI 兼容设备的信号引脚说明。

图 3.14 中引脚按功能来分主要有以下几类。

(1) 系统控制

CLK：PCI 时钟，上升沿有效。

$\overline{\text{RST}}$：复位信号。

(2) 接口控制

$\overline{\text{FRAME}}$：标志传输开始与结束。

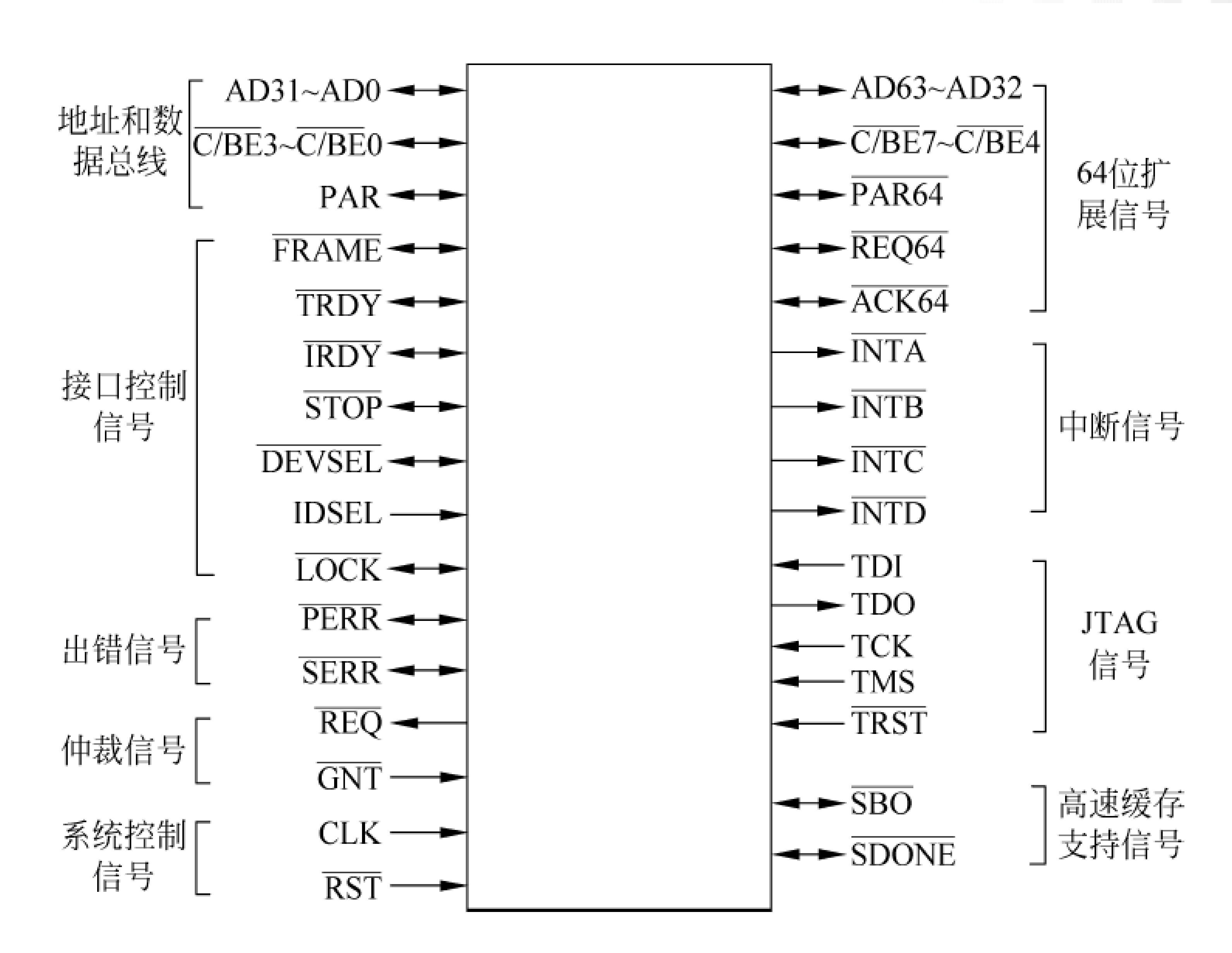


图 3.14　与 PCI 兼容设备的信号引脚

TRDY：Slave 可以传输数据的标志。

IRDY：Master 可以传输数据的标志。

STOP：Slave 主动结束传输数据的信号。

DEVSEL：当 Slave 发现自己被寻址时置低电平应答。

IDSEL：在即插即用系统启动时用于选中板卡的信号。

LOCK：锁定信号，当该信号有效时，一个动态操作可能需要多个传输来完成。

(3) 地址与数据总线

AD31～AD0：地址/数据分时复用总线。

C/BE3～C/BE0：命令/字节使能信号。

PAR：奇偶校验信号。

(4) 仲裁信号

REQ：Master 用来请求总线使用权的信号。

GNT：Arbiter 允许 Master 得到总线使用权的信号。

(5) 出错信号

PERR：数据奇偶校验出错。

SERR：系统奇偶校验出错。

(6) 中断信号

INTA～INTD：PCI 定义的一个中断向量对应一个设备，4 个以上中断向量对应一个多功能的设备或连接器。

(7) 其他可选信号

• 高速缓存支持信号

SBO：试探返回。当该信号有效时，关闭预示命中一个缓冲行。

SDONE：预示命中一个缓冲行。当它无效时，表明探测结果仍未确定；当它有效时，则

表明探测完成。

- 64 位扩展信号

AD63～AD32：地址/数据复用，提供 32 位附加位。

$\overline{\text{C/BE7}}$～$\overline{\text{C/BE4}}$：扩展高 32 位的总线命令和字节使能信号。

$\overline{\text{PAR64}}$：奇偶双字节校验，是 AD63～AD32 和$\overline{\text{C/BE7}}$～$\overline{\text{C/BE4}}$ 的校验位。

$\overline{\text{REQ64}}$：64 位传输请求。

$\overline{\text{ACK64}}$：标明从设备将采用 64 位传输。

- JTAG 信号

TDI：测试信号输入。

TDO：测试信号输出。

TCK：时钟测试。

TMS：测试模式选择。

$\overline{\text{TRST}}$：测试逻辑重置。

2）Compact PCI 总线简介

Compact PCI 总线简称 CPCI，又称紧凑型 PCI，是一种基于标准 PCI 总线的小巧而坚固的高性能总线技术。采用 CPCI 总线的板卡如图 3.15 所示。它是国际 PICMG（PCI Industrial Computer Manufacturers Group，PCI 工业计算机制造商联盟）协会于 1994 年提出来的一种总线接口标准。Compact PCI 所具有的开放性、高可靠性、可热插拔特性，使该技术除了可以广泛应用在通信、网络、计算机外，同时，也适用于实时系统控制、产业自动化、实时数据采集、军事系统等需要高速运算的领域，以及智能交通、航空航天、医疗器械、水利等模块化且需要极高可靠度、可长期使用的应用领域。

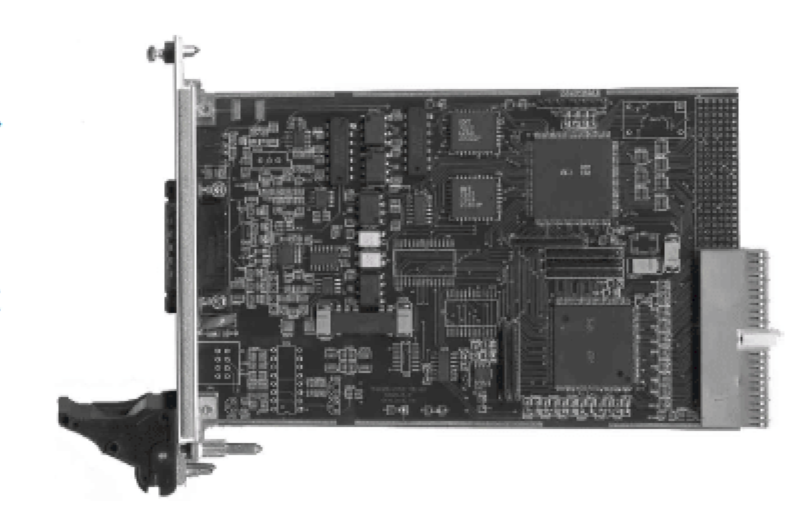


图 3.15 采用 CPCI 总线的板卡

Compact PCI 具有以下特点。

- 标准的 Eurocard（欧卡电路板规格）尺寸（根据 IEEE 1101.1 机械标准制订）。
- Compact PCI 使用符合 IEC-1076 国际标准高密度气密式针孔连接器，其 2mm 的脚具有低感抗和阻抗的特点，从而减少了高速 PCI 总线引起的信号反射。
- 具有较好的抗震性。传统的金手指连接器不能对系统中的外围设备板卡提供可靠而安全的支持，插于其中的板卡只能固定于一点，卡的顶端和底部也没有导轨支持，因此卡与槽的连接处容易在震动中接触不良。而 Compact PCI 卡顶端和底部均有导轨支持。前面板紧固装置将前面板与周围的机架安全地固定在一起。卡与槽的连接部分通过针孔连接器紧密地连接。因此即使在剧烈的冲击和震动场合，也能保证持久连接而不会接触不良。
- 通风性较好。传统的工业 PC 机箱内空气流动不畅，不能有效散热。Compact PCI 系统由于良好的机械设计带来通畅的散热途径，冷空气可以随意在板卡间流动，并将热量带走。集成在板卡底部的风扇系统也加速了散热进程。Compact PCI 系统极少出现散热方面的问题。
- 目前 Compact PCI 技术是国际先进的工业计算机技术。

3. PC/104 和 PC/104PLUS 总线

1）PC/104 总线

PC/104 总线是嵌入式 PC 所用的小型化的总线标准，是一种专门为嵌入式控制而定义的工业控制总线。有两个总线插头，共有 104 个引脚，这也是 PC/104 名称的由来。PC/104 有两个版本，8 位和 16 位，分别与 PC 和 PC/AT(ISA)相对应。本质上就是尺寸缩小为 3.8 英寸、3.9 英寸的 ISA 总线板卡。总线以“针”和“孔”形式层叠连接，这种层叠封装有极好的抗震性。4mA 总线驱动电流即可使模块正常工作，每个模块 1～2W 能耗。最高工作频率为 8MHz，数据传输速率达到 8MB/s，地址线 24 条，可寻访 16MB 地址单元。

104 根信号线分为 5 类：地址线、数据线、控制线、时钟线和电源线。基于 PC/104 总线的板卡如图 3.16 所示。

2）PC/104PLUS 总线

PC/104PLUS 总线与 PCI 总线相对应，采用单列三排 120 个管脚。有效信号线和控制线与 PCI 总线完全兼容，可以连接高速外接设备，包括了 PCI 规范 2.1 版要求的所有信号。为了向下兼容，PC/104PLUS 保持了 PC/104 的所有特性。基于 PC/104PLUS 总线的板卡如图 3.17 所示。



图 3.16　基于 PC/104 总线的板卡

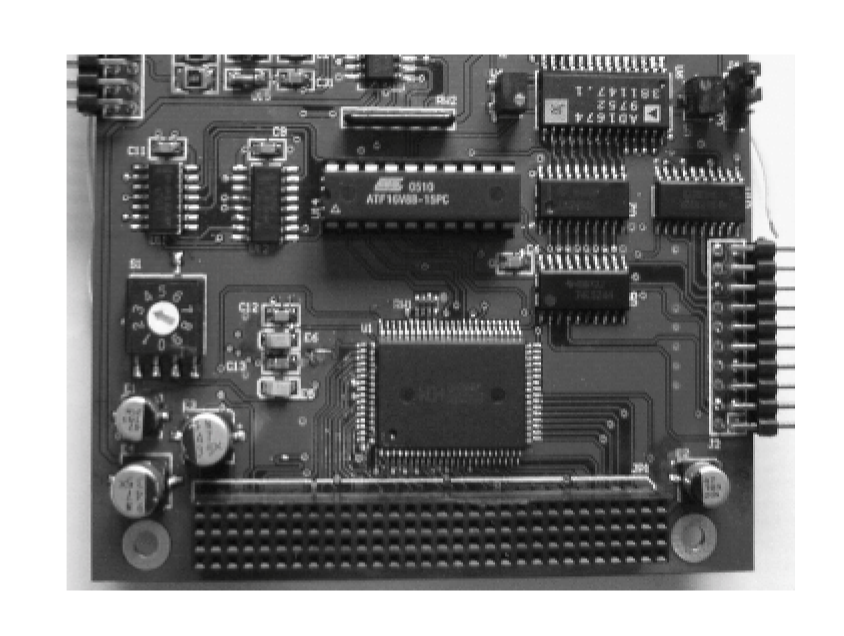


图 3.17　基于 PC/104PLUS 总线的板卡

3.2.3　外总线

1. STD 总线

STD 总线属外总线，是美国普洛公司于 1978 年推出，适用于工业控制机的特点和要求而设计的 56 根线的总线，如图 3.18 所示。采用公共母板结构，板上安装若干个插座，插座对应引脚都是连到同一根总线信号线上，总线信号符合 STD 规范，各功能模块（如 CPU 模块、存储器模块、图形显示模块、A/D 模块、D/A 模块、开关量 I/O 模块等）都按标准尺寸制作，可插入任意插座，从而组建成不同规模的计算机系统。STD 总线的 16 位总线性能满足嵌入式和实时性应用要求。在 56 根总线中，有 6 根逻辑电源线、4 根辅助电源线、8 根数据总线、16 根地址总线和 22 根控制总线。

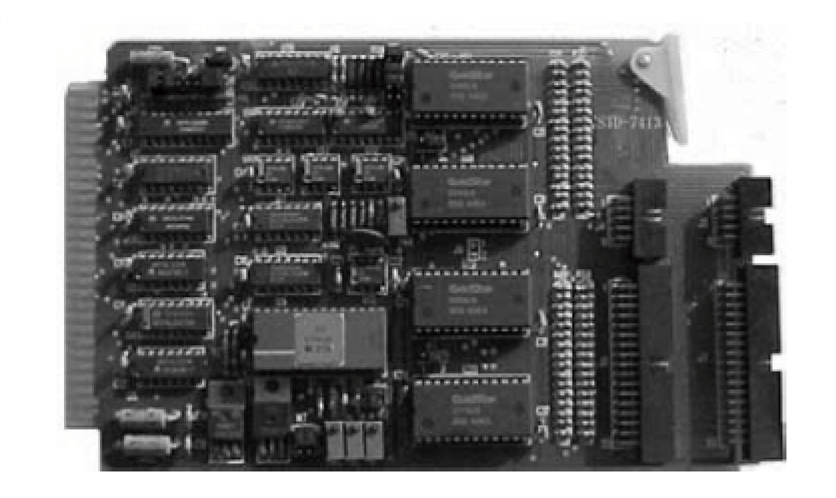


图 3.18　采用 STD 总线的板卡

2. IEEE-488 总线

IEEE-488 总线是并行总线接口标准，使用 24 芯连接器，其中有 8 条双向数据线、3 条字节传送控制线、5 条接口管理线及 8 条地线。如微计算机、数字电压表、数码显示器等设备及其他仪器仪表均可用 IEEE-488 总线装配起来。它是按照位并行、字节串行双向异步方式传输信号，连接方式为总线方式，仪器设备直接并联于总线上而不需中介单元，总线上最多可连接 15 台设备，可以采用串行或星形方式连接。最大传输距离为 20m，信号传输速度一般为 500KB/s，最大传输速度为 1MB/s。如图 3.19 为采用 IEEE-488 总线的板卡。



图 3.19 IEEE-488 总线板卡

连接至 IEEE-488 总线上的微型计算机及其他设备，接口一般采用专用的接口芯片来实现，每一个设备可按如下三种方式工作。

(1) 听者方式，即从数据总线上接收数据，一个系统在同一时刻可以有两个以上的“听”者，可以充当听者的设备有微型计算机、打印机、绘图仪等。

(2) 讲者方式，即向总线发送数据，一个系统可以有两个以上的“讲”者，但任一时刻，只能有一个讲者在工作，可以充当讲者的设备有微型计算机、数字电压表、频谱分析仪等。

(3) 控者方式，即向其他设备发布命令，例如对其他设备寻址，或允许讲者使用总线等。控制者常由微型计算机担任，一个系统可以有多个控制者，但每一时刻只能有一个发出控制命令。

3. 串行外部总线

以上所述总线方式主要采用并行通信方式，即数据的各位同时传送。并行方式传输数据速度快，但占用的通信线多，传输数据的可靠性随距离的增加而下降，只适用于近距离的数据传送。而串行通信是指在单根数据线上将数据一位一位地依次传送。在远距离数据通信中，一般采用串行通信方式，它具有占用通信线少、成本低等优点。

串行通信又分为同步和异步两种通信方式。所谓同步通信，是指数据传送是以数据块(一组字符)为单位，字符与字符之间、字符内部的位与位之间都同步。同步串行通信的特点可以概括如下。

(1) 以数据块为单位传送信息。

(2) 在一个数据块(信息帧)内,字符与字符间无间隔。

(3) 接收时钟与发送时钟严格同步。

同步串行通信的数据格式如图 3.20 所示,每个数据块(信息帧)由 3 个部分组成:①两个同步字符作为一个数据块(信息帧)的起始标志;②n 个连续传送的数据;③两个字节循环冗余校验码(CRC)。

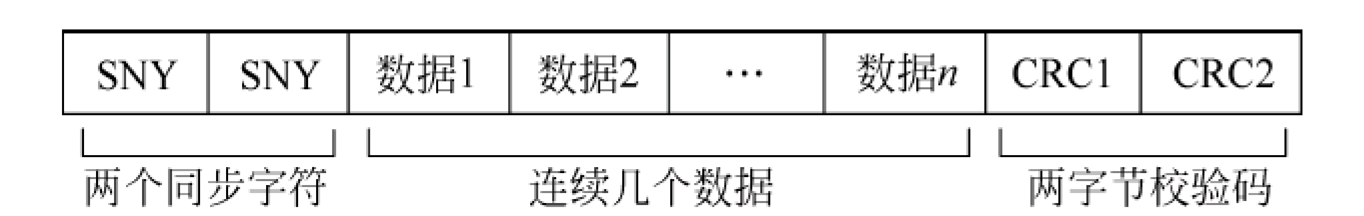


图 3.20　串行通信的数据格式

异步通信是指数据传送以字符为单位,字符与字符间的传送是完全异步的,位与位之间的传送基本上是同步的。简单地说就是:字符间异步,字符内部各位同步。其特点可以概括为以下几项。

(1) 以字符为单位传送信息。

(2) 相邻两字符间的间隔是任意长。

(3) 因为一个字符中的比特位长度有限,所以需要的接收时钟和发送时钟只要相近就可以。

异步串行通信的数据格式如图 3.21 所示,每个字符(每帧信息)由 4 个部分组成:①1 位起始位,规定为低电平 0;②5～8 位数据位,即要传送的有效信息;③1 位奇偶校验位;④1、2 位停止位,规定为高电平 1。

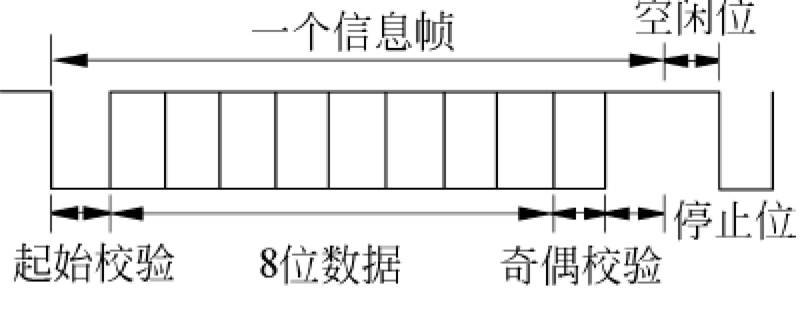


图 3.21　异步串行通信的数据格式

根据数据传送方式,串行总线又分为以下三种方式。

- 单工方式:只允许数据按固定的方向传输。
- 半双工方式:采用同一根数据传输线,允许数据分时在两个方向传输,但不能同时进行双向传送。
- 全双工方式:允许数据同时双向传送。

另外,串行数据传输的线路通常有平衡方式和不平衡方式之分。所谓平衡方式就是用双绞线传输信号,信号在双绞线中自成回路,而不通过地线,接收器是用双端差动方式接收信号的。而不平衡方式采用单线传输信号,以地线作为信号的回路,接收器用单线接收信号。

下面介绍几种常用的串行外部总线。

1) RS-232 总线

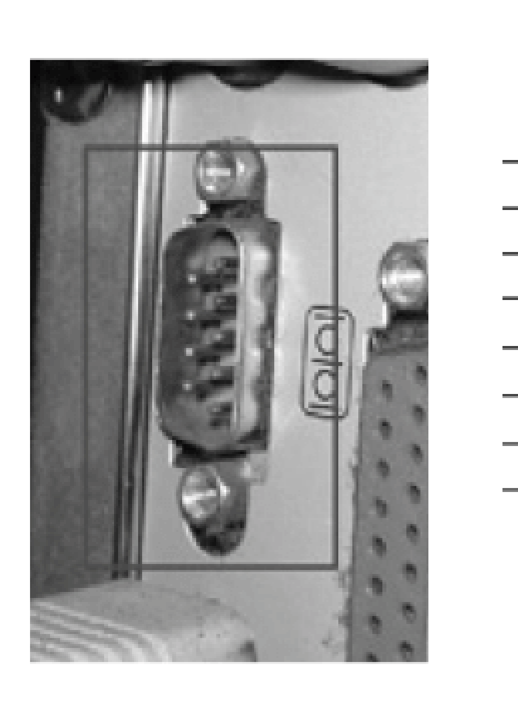
RS-232 是美国电子工业协会 EIA(Electronic Industry Association)制定的一种串行物理接口标准。RS 是英文"推荐标准"(Recommended Standard)的缩写,232 为标识号,如果是 RS-232C,则 C 表示修改次数。通常以 9 个引脚(DB9)或是 25 个引脚(DB25)的形态出现,一般计算机上会有两组 RS232 接口,分别称为 COM1 和 COM2。DB25 的串口用到的管脚一般只有 2 脚(RXD)、3 脚(TXD)、7 脚(GND)这三个,随着设备的不断改进,现在

DB25 针很少看到了,一般是 DB9 的接口,DB9 所用到的管脚是 2 脚(TXD)、3 脚(RXD)、5 脚(GND)。因此也把 RS2-32 接口叫做 DB9。RS-232 接口及在 Proteus 软件中的电路符号如图 3.22 所示。

9 针 RS-232 串口引脚定义如下:

- 1 DCD 载波侦测(Data Carrier Detect);
- 2 RXD 接收数据(Receive);
- 3 TXD 发送数据(Transmit);
- 4 DTR 数据终端准备(Data Terminal Ready);
- 5 GND 地线(Ground);
- 6 DSR 数据准备好(Data Set Ready);
- 7 RTS 请求发送(Request To Send);
- 8 CTS 清除发送(Clear To Send);
- 9 RI 振铃指示(Ring Indicator)。

RS-232 采取不平衡传输方式,即所谓单端通信,其收发端的数据信号都是相对于地信号的,所以其共模抑制能力差;再加上双绞线的分布电容,如 RS-232C 标准规定,驱动器允许有 2500pF 的电容负载,通信距离将受此电容限制,当采用 150pF/m 的通信电缆时,最大通信距离为 15m,若每米电缆的电容量减小,通信距离可以增加。且其只支持点对点通信,不能实现联网功能。若要进行联网,需通过一个 RS-232/RS-485 转换器将 RS-232 总线转换成 RS-485 总线来实现。图 3.23 为 RS-232/RS-485 转换器实物图。



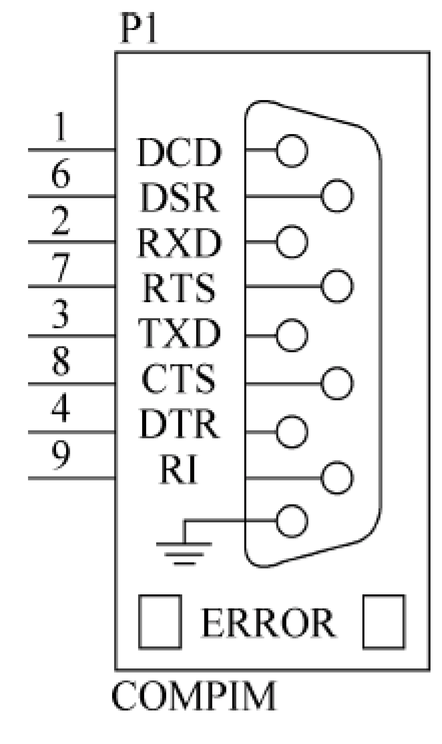


图 3.22 RS-232 接口及在 Proteus 软件中的电路符号

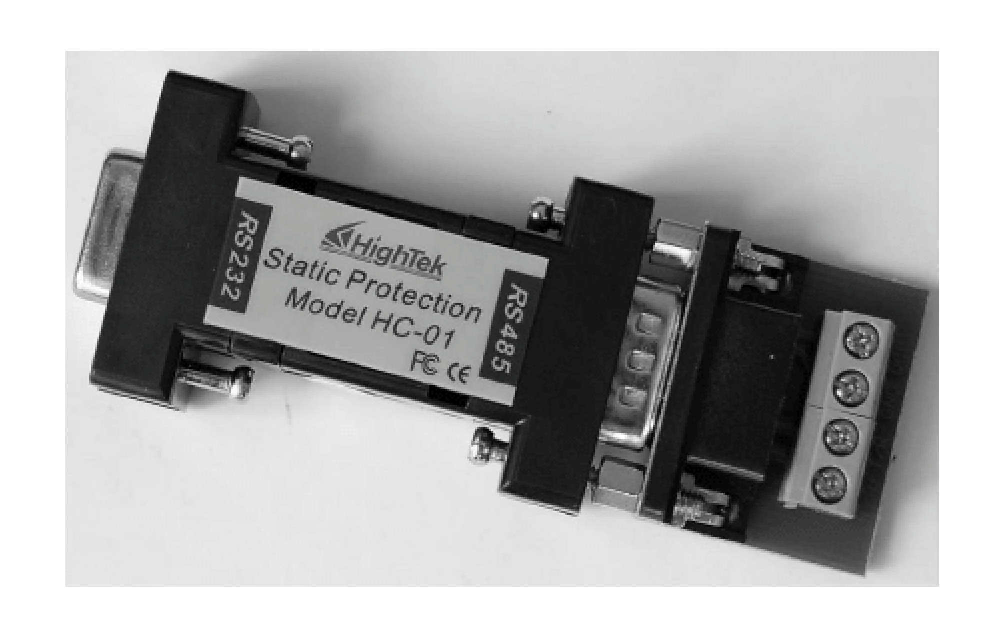


图 3.23 RS-232/RS-485 转换器

RS-232 对电器特性、逻辑电平和各种信号线功能都作了规定。在 TXD 和 RXD 上逻辑 1 为 $-3\sim-15$V;逻辑 0 为 $+3\sim+15$V;在 RTS、CTS、DSR、DTR 和 DCD 等控制线上,信号有效(接通,ON 状态,正电压)为 $+3\sim+15$V,信号无效(断开,OFF 状态,负电压)为 $-3\sim-15$V。

综上所述,RS-232 的特点有以下几个方面。

(1) 传输速率较低。

(2) 传输距离有限。

(3) 抗噪声干扰性弱。

(4) 接口的信号电平值较高，易损坏接口电路的芯片，又因为与 TTL 电平不兼容故需使用电平转换电路方能与 TTL 电路连接。

(5) 使用线路少，成本低。

2) RS-422 总线

RS-422 总线由 RS-232 发展而来，为改进 RS-232 通信距离短、传输速度低的缺点。RS-422 总线采用 4 线制，采取平衡传输方式(差分传输方式)，将传输速率提高到 10Mb/s，并允许在一条平衡总线上连接最多 10 个接收器，是一种单机发送、多机接收的平衡传输规范。因为采用了差动接收方法，接收到的信号是两个输入端的电位差，所以信号的电平幅度不必抬高，用 TTL 电平即可；同时，发送端和接收端不共地，所以具有很强的抗干扰能力，直接相连时，传输距离可达 1200m。图 3.24 是典型的 RS-422 四线接口，实际上还有一根信号地线，共 5 根线。且 RS-422 需要接终接电阻，要求其阻值约等于传输电缆的特性阻抗。在短距离传输(一般在 300 米以下)时可不需终接电阻。终接电阻接在传输电缆的最远端。

由于 RS-422 总线的四线接口采用单独的发送和接收通道，因此支持全双工方式通信。

综上，RS-422 总线的主要特点包括以下几项。

(1) 支持点对多点的双向通信，可实现一个主设备与多个从设备通信，但从设备之间不能通信。

(2) 传输距离长，最大距离一千多米。

(3) 抗干扰能力强。

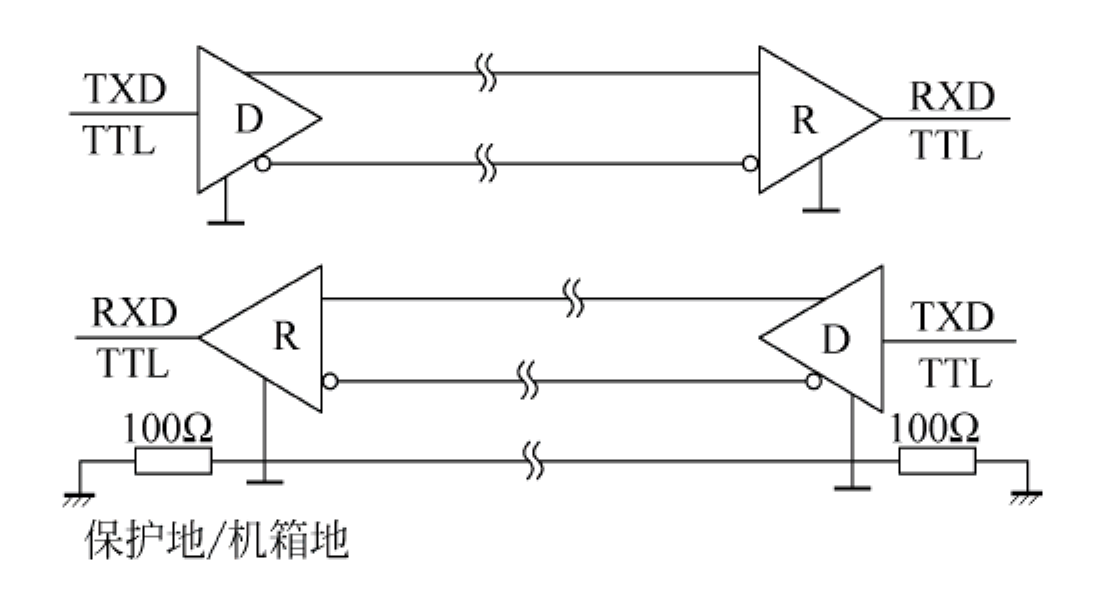


图 3.24　RS-422 四线接口

3) RS-485 总线

为了在距离较远的情况下进一步节省通信电缆的费用，在 RS-422 总线的基础上推出了 RS-485 总线。由于 RS-485 是从 RS-422 基础上发展而来的，所以 RS-485 许多电气规定与 RS-422 相仿，如都采用平衡传输方式，都需要在传输线上连接终接电阻等。RS-485 可以采用二线与四线方式。二线制可实现真正的多点双向通信，与 RS-422 总线的接口基本没有区别，只是在发送端增加了使能控制，属于半双工方式通信，RS-488 二线接口如图 3.25 所示；采用四线连接时，与 RS-422 一样只能实现点对多的通信，即只能有一个主(Master)设备，其余为从设备，但它比 RS-422 有所改进，无论四线还是二线连接方式总线上最多都可接到 32 个设备。

RS-485 与 RS-422 一样，其最大传输距离约为 1219m，最大传输速率为 10Mb/s。平衡双绞线的长度与传输速率成反比，在 100Kb/s 速率以下，才可能使用规定最长的电缆长度，只有在很短的距离下才能获得最高速率传输。一般 100m 长双绞线最大传输速率仅为 1Mb/s。

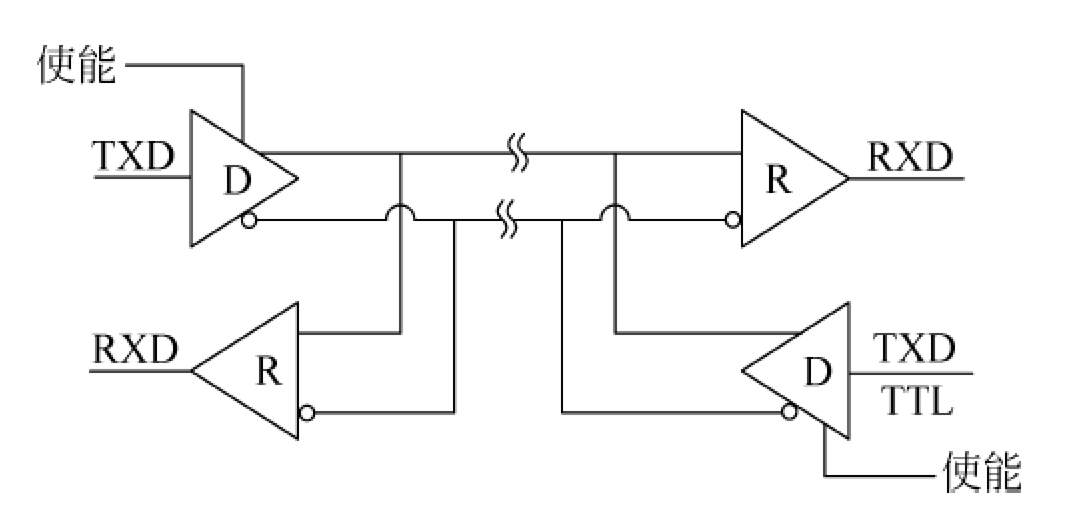


图 3.25 RS-485 二线接口

RS-485 的主要特点包括以下几项。

(1) 支持多点到多点的通信。

(2) 采用半双工工作方式。

(3) 具有抑制共模干扰的能力。

(4) 传输距离远。

RS-232、RS-422、RS-485 接口比较如表 3.1 所示。

表 3.1 RS-232、RS-422、RS-485 接口比较

接口名称	RS-232	RS-422	RS-485
传输方式	电平	差分	差分
最大传输速率	20Kb/s(15m)	100Kb/s(1200m)	100Kb/s(1200m)
工作方式	全双工	全双工	半双工
连接方式	点到点	点到多点	多点到多点

4) 通用串行总线 USB

USB(Universal Serial Bus)是由 Compaq(康柏)、HP(惠普)、Intel(英特尔)、Lucent(朗讯)、Microsoft(微软)、NEC(日本电气)和 Philips(飞利浦)七家公司联合推出的新一代标准接口总线。该总线是一种连接外围设备的机外总线,最多可连接 127 个设备,为微机系统扩充和配置外部设备提供了方便。USB 采用四线电缆,其中两根是用来传送数据的串行通道,另两根为外部设备提供电源,对于高速且需要高带宽的外设,USB 以全速 12Mb/s 的传输速率来传输数据;对于低速外设,USB 则以 1.5Mb/s 的传输速率来传输数据。USB 总线会根据外设情况在两种传输模式中自动地动态转换。USB 电缆如图 3.26 所示。

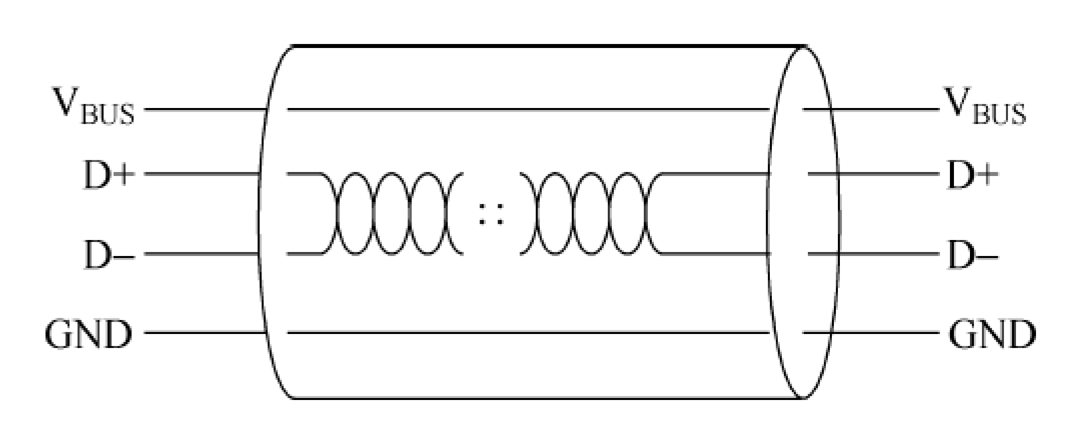


图 3.26 USB 电缆

USB 总线的主要优点有以下几项。

(1) 可以进行热插拔。

(2) 系统总线供电,低功耗设备无需外接电源。

(3) 扩展容易,可以连接多个外部设备,最多可达 127 个。

(4) 高速数据传输,USB 1.1 是 12Mb/s,USB 2.0 高达 480Mb/s。

USB 总线的主要缺点有以下几项。

(1) 供电能力：如果外设的供电电流大于 500mA 时,设备必须外接电源。

(2) 传输距离：USB 总线的连线长度最大为 5m。

3.3　数字量输入输出通道

在计算机控制系统中,数字信号是需要处理的一类基本的输入/输出信号,主要包括电平式信号和触点式信号,它们以二进制的逻辑“1”、“0”或高电平、低电平出现。基本的输入输出数字量包括以下几项。

- 开关的闭合与断开。
- 指示灯的亮与灭。
- 继电器或接触器的吸合与释放。
- 电动机的供电与断电。
- 电磁阀等阀门的打开与闭合。
- 晶闸管的通与断。
- 仪器仪表的 BCD 码等。

把数字量输入通道或数字量输出通道设计在一块板卡上，就称为 DI 板卡或 DO 板卡，也可统称为数字量 I/O 板卡。图 3.27 即为含有 DI 通道和 DO 通道的 PC 总线数字量 I/O 板卡的结构框图。

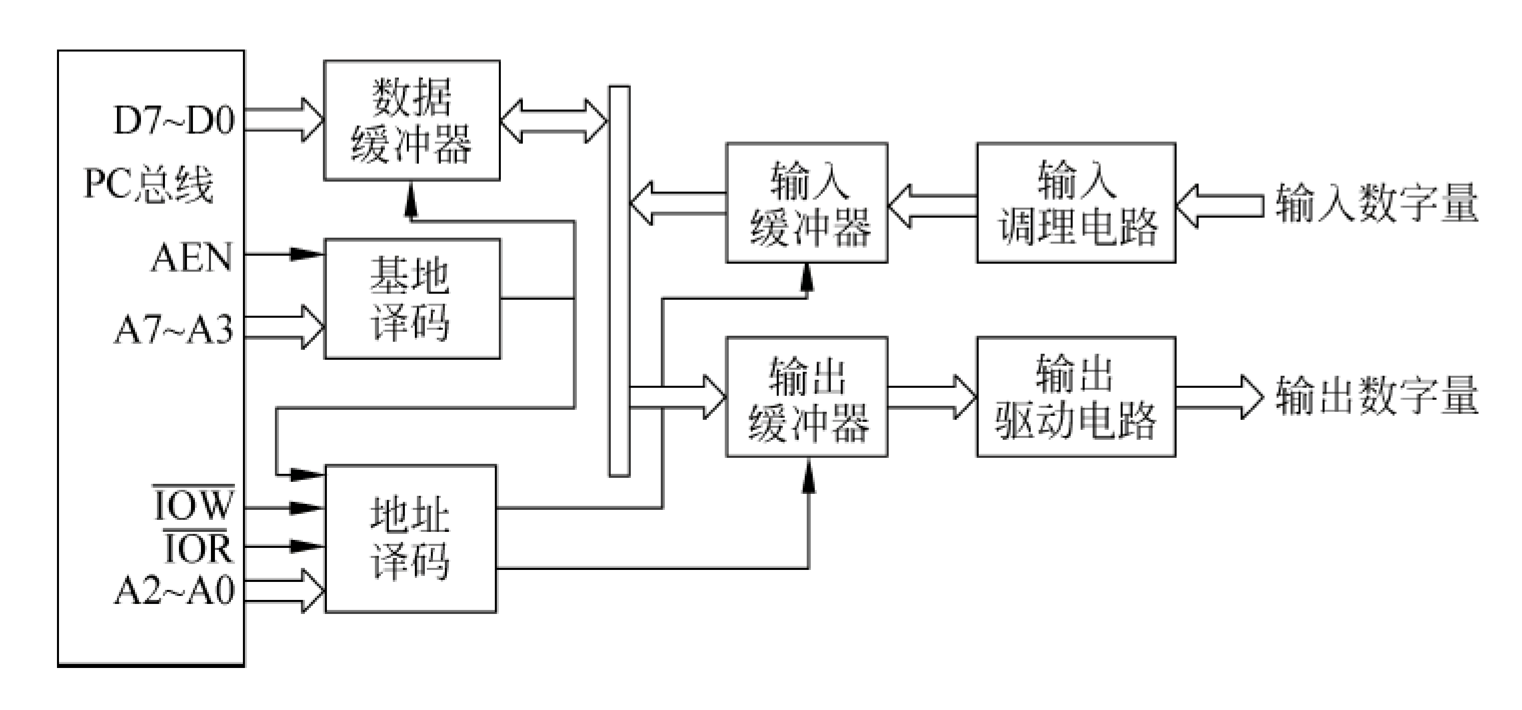


图 3.27　数字量 I/O 板卡的结构框图

3.3.1　数字量输入通道

数字量输入通道是把外界生产过程的开关状态信号或数字信号送至计算机进行处理，简称 DI(Digital Input)通道。主要由输入调理电路、输入缓冲器、输入地址译码电路等组成,如图 3.28 所示。

1. 地址译码电路

在计算机控制系统中,所有的 I/O 端口或存储设备都以地址来相互区分,根据访问 I/O

端口或存储器指令中的地址信息，地址译码电路产生相应的寻址信号，以选中对应的 I/O 端口或存储器。这些地址称为 I/O 端口地址或存储器地址。通常一个 I/O 端口包含数据端口、命令端口和状态端口。

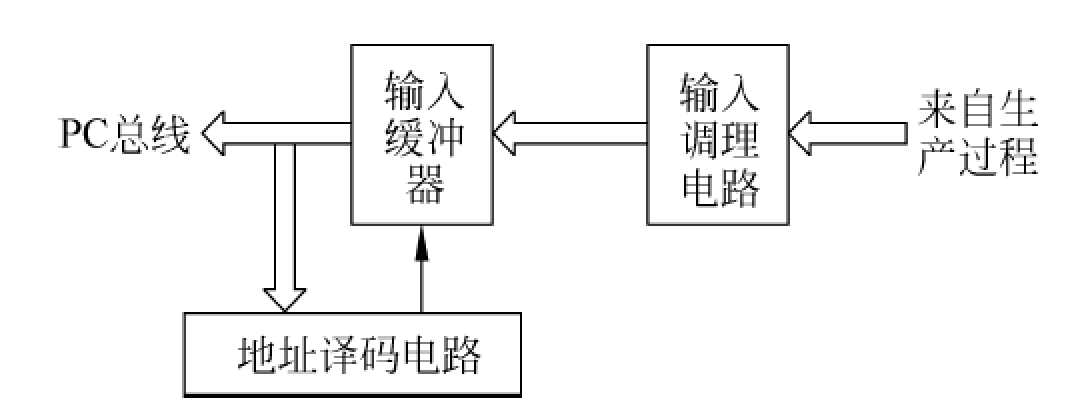


图 3.28 数字量输入通道组成

端口地址的设置方法一般有两种：统一编址和独立编址。统一编址是把 I/O 端口的端口地址归入存储器寻址地址空间范围内，也称为存储器映像编址。因此在其指令系统中，没有专门的 I/O 指令，共用访问存储器的指令，即 CPU 访问一个端口的操作与访问内存的操作一样。独立编址是 I/O 接口地址和存储器地址分别独立编址，采用专门的 I/O 指令来对接口地址进行操作。这样，存储器地址和 I/O 接口地址可以重叠。然而由于两者需采用不同的指令进行读写操作，所以不会由于地址相同而混淆。这种编址方式的优点是不占用存储器地址，因而不会减少存储器容量。由于有专门的 IN 和 OUT 指令，因此比用存储器读写指令执行速度快。但这种方式的缺点是在硬件电路上要对这两种存取进行区别。另外，专门的 I/O 指令功能简单，要完成某些操作还须和某些指令配合。

地址译码通常采用全译码电路、部分译码电路、线选译码电路等不同的方式。全译码电路是指将高位地址信号经译码后作片选信号，低位地址信号用于 I/O 端口或存储器的寻址，也就是说，所有的地址信号都参与编址，不会发生地址重叠的现象。部分译码电路将 I/O 端口或存储器所用的低位地址线外的高位地址线中的部分地址信号作为译码电路的输入信号，这样会发生地址的重叠现象。而线选译码电路是直接利用地址线作为端口或存储器的片选信号，其使用的地址是不连续的，而且在使用时必须防止两条地址线同时为低电平，否则将使系统同时选中两个对象而无法正常工作。

如图 3.29 所示数字量输入接口电路中，左侧为 ISA 总线，包括地址总线、数据总线、命令总线等；74HCT688(8 位数据比较器)为板地址译码电路，采用了开关选择译码方法，当地址总线 A9～A4 上送来地址信号(Q 口)与拨码开关 DSW1 所选输入数字信号(P 口)相一致时，74HCT688 输出有效的低电平信号 BS，用于选通输入数据缓冲器 74HCT244 和译码器 74HCT138；74HCT138 为板内 I/O 端口地址译码电路，它根据地址总线所送来的地址信号 A2～A0，对应选中 Y7～Y0，从而进行端口的选择。

例如当 CPU 执行读命令时($\overline{\text{IOR}}$有效)，如果地址信号 A2～A0 为 000，则 Y0 低电平有效，端口 1 被选中，进而数据(ID0～ID7)从端口 1 输入；在读信号和 BS 信号(由地址 A9～A4 所确定)作用下，选通输入数据缓冲器 74HCT244，经 74HCT244 将数字信号送至系统数据总线，完成一次数字量输入的操作。

另外，为了对上述电路进行动态仿真，图 3.29 是采用 Proteus 软件的 ISIS 进行的原理图设计，并加入了输入输出的逻辑状态显示，具体设计方法在第 4 章详细介绍。

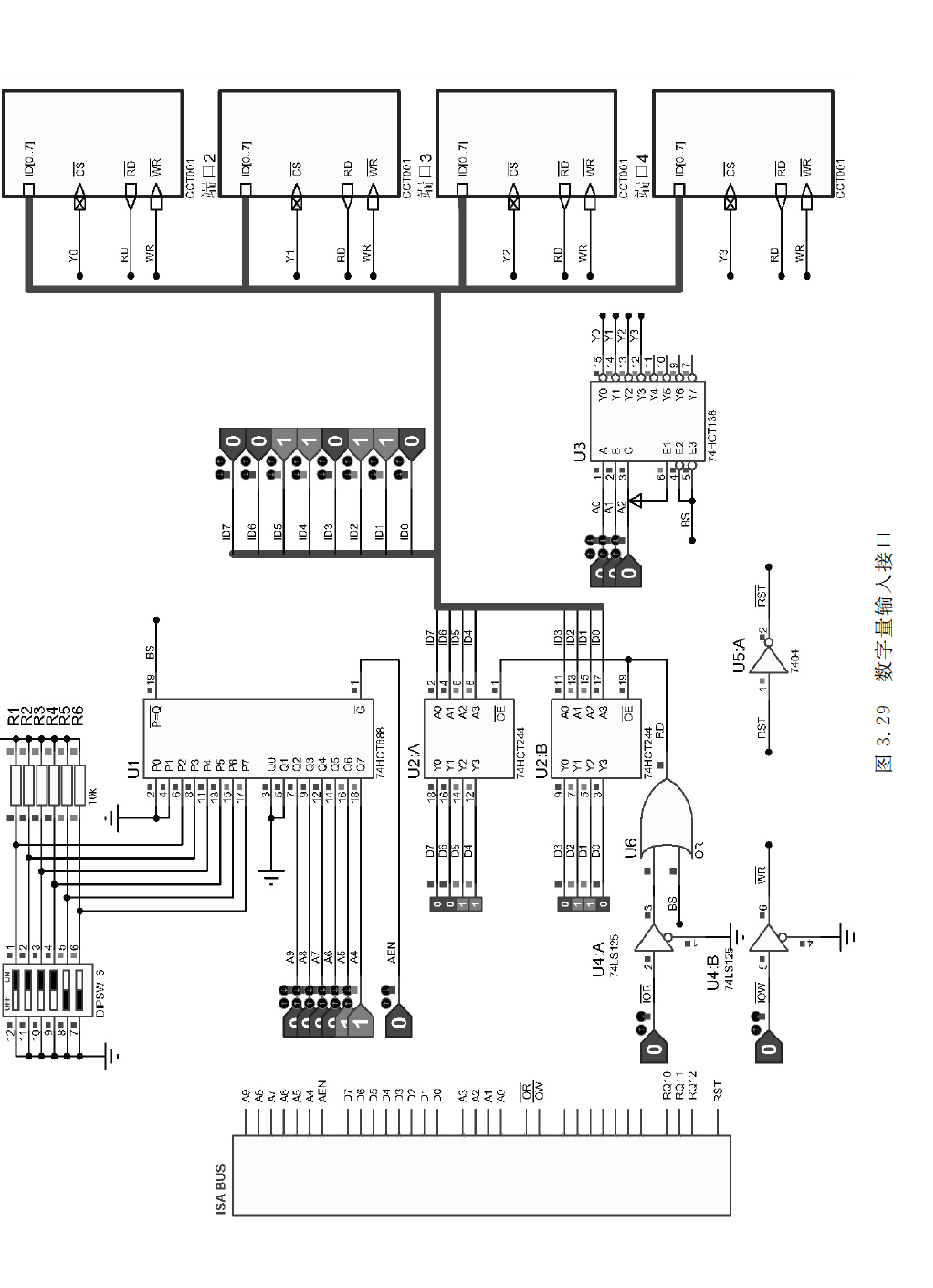


图 3.29 数字量输入接口

可见,译码不仅仅只与地址信号有关,与控制信号也有关联。例如上例中 I/O 译码电路是把地址信号 A9～A0、命令信号$\overline{\text{IOR}}$(或$\overline{\text{IOW}}$)、AEN(为 0 时表示非 DMA 传送)进行组合,产生端口的选择信号。另外,还会用到像$\overline{\text{IO16}}$来控制对 8 位还是 16 位操作,用$\overline{\text{SBHE}}$来控制端口的奇偶地址等。

Turbo C 提供了专门对 I/O 端口进行输入输出操作的几个函数,分别是:

- int inp(int portid)。
- int inport(int portid)。
- int inportb(int portid)。
- void outp(int portid, int value)。
- void outport(int portid,int value)。
- void outportb(int portid,int value)。

这些函数的原型在 dos. h 中。其中,inp 是等价于 inport 的一个宏,outp 是等价于 outport 的一个宏,因而我们仅需了解其他几个函数就可以了。

inport 函数从指定的接口地址 portid 中读入一个字(即 16 位二进制数),而 inportb 则从指定的接口地址 portid 中读入一个字节(8 位二进制数)。如 I/O 接口的寻址方式中所述,当执行这两个函数后,它们均返回各自从接口地址所对应的输入设备中得到的 16 位或 8 位二进制数。outport 函数把一个 16 位二进制数 value 发送到口地址为 portid 的接口中去。对于 PC,实际上是将低字节数送到口地址为 portid 接口中去,将高字节送到口地址为 portid+1 的接口中去。outportb 函数是将一个字节的数 value 送到口地址为 portid 的接口中去。

2. 输入缓冲器

输入缓冲器的作用是将外部端口送来的数据暂时存放,以配合处理器读取,对高速工作的 CPU 与慢速工作的外设起协调和缓冲作用,以实现数据传送的同步。由于缓冲器接在数据总线上,故必须具有三态输出功能。例如,图 3.29 所示数字量输入接口电路中 8 同相三态缓冲器线驱动器 74HCT244 可用来隔离输入和输出线路,起到数据缓冲作用;三态输出 4 总线缓冲器 74LS125 可对读、写信号进行缓冲。另外,I/O 缓冲功能也可以用可编程接口芯片如 8255A 构成,或者用 74LS240、74LS244、74LS373、74LS273 等芯片实现。

3. 输入调理电路

输入调理电路主要针对外部输入计算机的状态信号可能是电压、电流、开关的触点状态等,容易引起瞬时高压、过高压、接触抖动等现象,因此,这些信号输入到计算机中之前,需经过转换、保护、滤波、隔离等措施,将其转变成计算机能正确接收的逻辑信号,这些功能即称为信号调理。

典型的开关量输入信号调理电路如图 3.30 所示。点划线右边是由开关 S1 与电源组成的外部电路,图 3.30(a)是直流输入电路,图 3.30(b)是交流输入电路。交流输入电路比直流输入电路多一个降压电容和整流桥模块。开关 S1 的状态经 RC(R1 和 C1)滤波、稳压管 D1 箝位保护、电阻 R2 限流、二极管 D2 防止反极性电压输入以及光耦隔离等措施处理后送

至输入缓冲器，主机通过执行输入指令便可读取开关 S 的状态。如开关 S1 闭合时，输入回路有电流流过，光耦中的发光管发光，光敏管导通，数据线上为低电平，即输入信号为“0”；反之，开关 S 断开，输入信号为“1”。

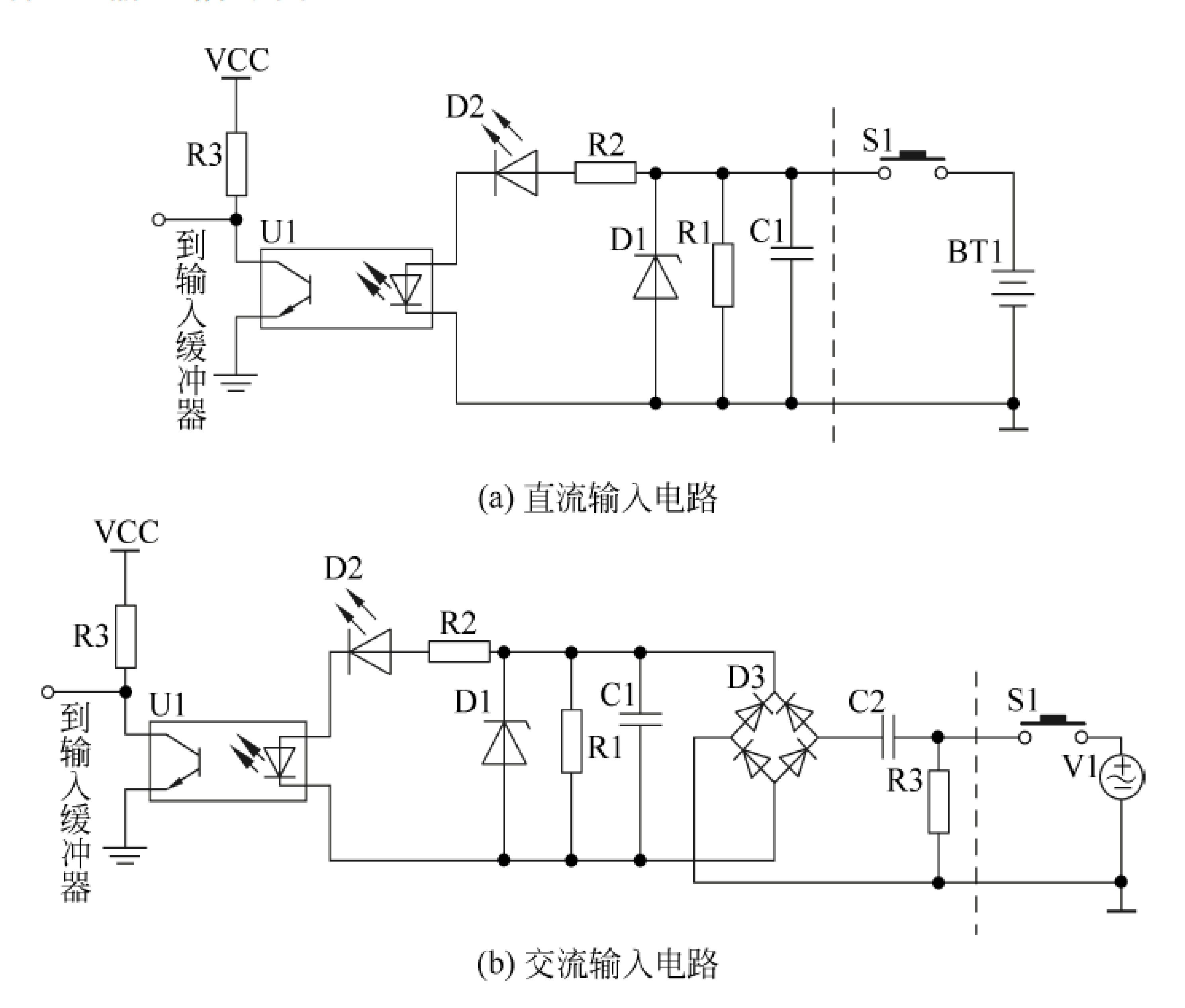


图 3.30　典型开关量输入信号调理电路

其中光电耦合器是工程中常用的输入隔离器件，它可以起到阻断信号源与信号接收方的电气连接，通过光信号传递信息，这样可以有效地阻断电气干扰。发光源的引脚为输入端，受光器的引脚为输出端，常见的发光源为发光二极管，受光器为光敏二极管、光敏三极管等。典型的光电耦合隔离电路有数字量同相传递与数字量反相传递两种，如图 3.31 所示。当光耦输出端从光敏管集电极引出，输入数字量为低电平“0”时，发光管导通且发光，使得光敏管导通，输出端接地而获得低电平“0”；当数据线为高电平“1”，发光管截止不发光，则光敏管也截止使输出端从电源处获得高电平“1”，因此属同相传递。同理可分析反相传递。

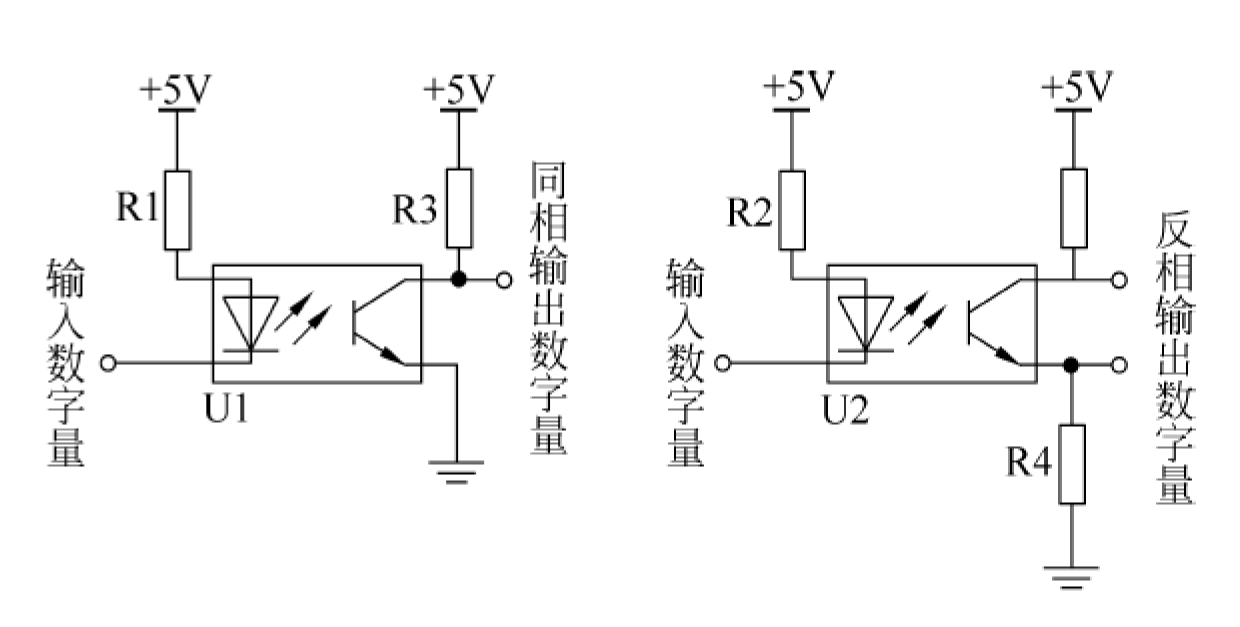


图 3.31　光耦的数字量同相传递与数字量反相传递

3.3.2 数字量输出通道

数字量输出通道是把计算机或微处理器送出的数字信号(或开关信号)传送给开关器件,如指示灯、继电器等,控制它们的闭合、断开、亮灭等,简称DO(Digital Output)通道。DO通道主要由输出锁存器、输出驱动电路、输出口地址译码电路等组成,如图3.32所示。其中地址译码电路与输入通道类似,这里不再赘述。

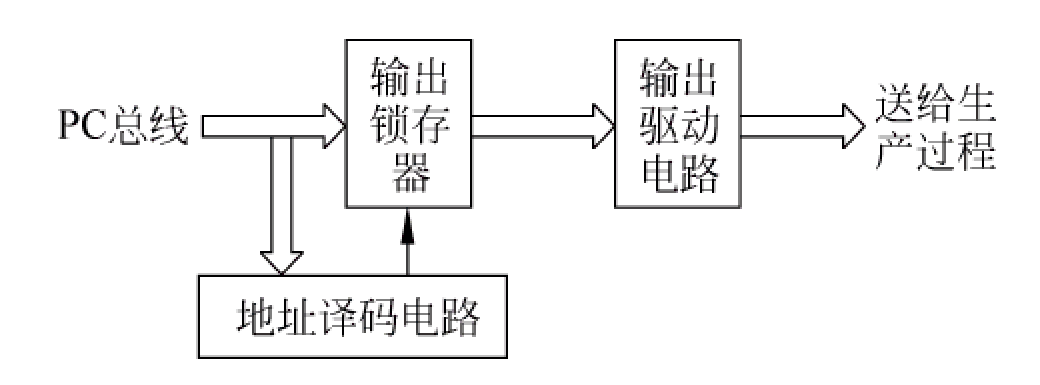


图3.32 数字量输出通道组成

1. 输出锁存器

当对生产过程进行控制时,输出的控制信号需要进行保持,因此,需进行输出数据的锁存。锁存器可以在有锁存信号时将输入的状态保存到输出,直到下一个锁存信号。锁存器的最主要作用是缓存,其次完成高速的控制与慢速的外设不同步的问题,再次是解决驱动的问题。例如图3.33所示为由74273作为输出锁存器的数字量输出接口电路。

2. 输出驱动电路

数字量输出通道的任务是把计算机输出的微弱数字信号转换成能对生产过程进行控制的数字驱动信号。根据现场负荷的不同,如指示灯、继电器、接触器、电机、阀门等,可以选用不同的功率放大器件组成不同的输出驱动通道。常用的有三极管输出驱动电路、继电器输出驱动电路、晶闸管输出驱动电路、固态继电器输出驱动电路等。

1) 三极管输出驱动电路

对于低压、小电流的开关量,用功率三极管就可作开关驱动组件,其输出电流就是输入电流与三极管增益的乘积。当驱动电流只有十几毫安或几十毫安时,只需采用一个普通的功率三极管就能构成驱动电路,如图3.34为在Proteus软件中所搭建功率三极管输出驱动继电器电路的仿真情况。

当驱动电流需要达到几百毫安时,如驱动中功率继电器、电磁开关等装置,输出电路必须采取多级放大或提高三极管增益的办法。达林顿阵列驱动器是由多组两个三极管组成的达林顿复合管构成,它具有高输入阻抗、高增益、输出功率大及保护措施完善的特点,同时也适用于计算机控制系统中的多路负荷。如图3.35为达林顿阵列ULN2003A的引脚定义及内部原理图,其电流增益高(大于1000),带负载能力强(输出电流大于500mA)。

图3.36为达林顿阵列驱动大功率继电器的其中一路驱动电路,当CPU数据线D_i输出数字“0”即低电平时,经7406反相锁存器变为高电平,使达林顿复合管导通,产生集电极电流足以驱动负载线圈,而且利用复合管内的保护二极管(输出连接端为第9引脚)构成了负荷线圈断电时产生的反向电动势的泄流回路。

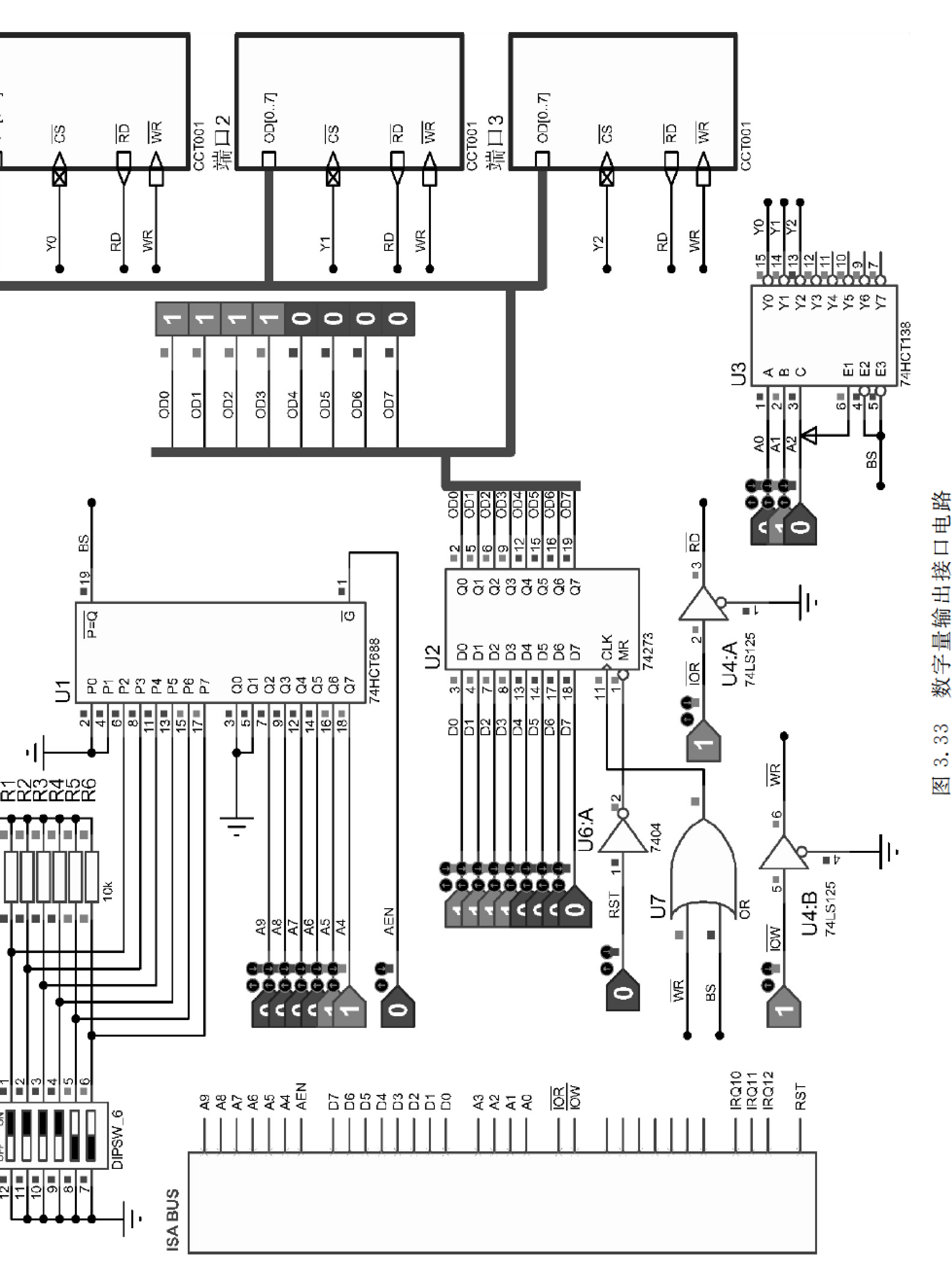


图 3.33　数字量输出接口电路

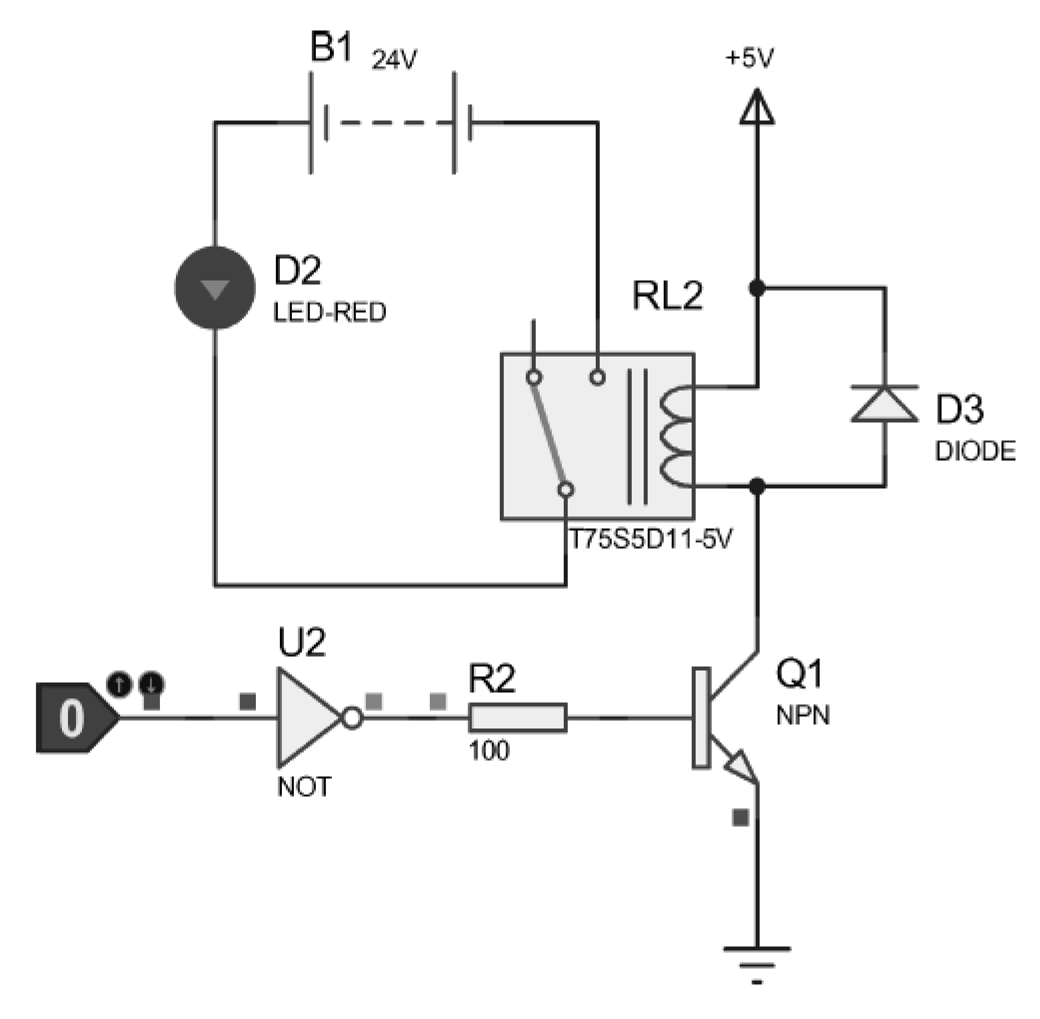


图 3.34　功率三极管输出驱动继电器

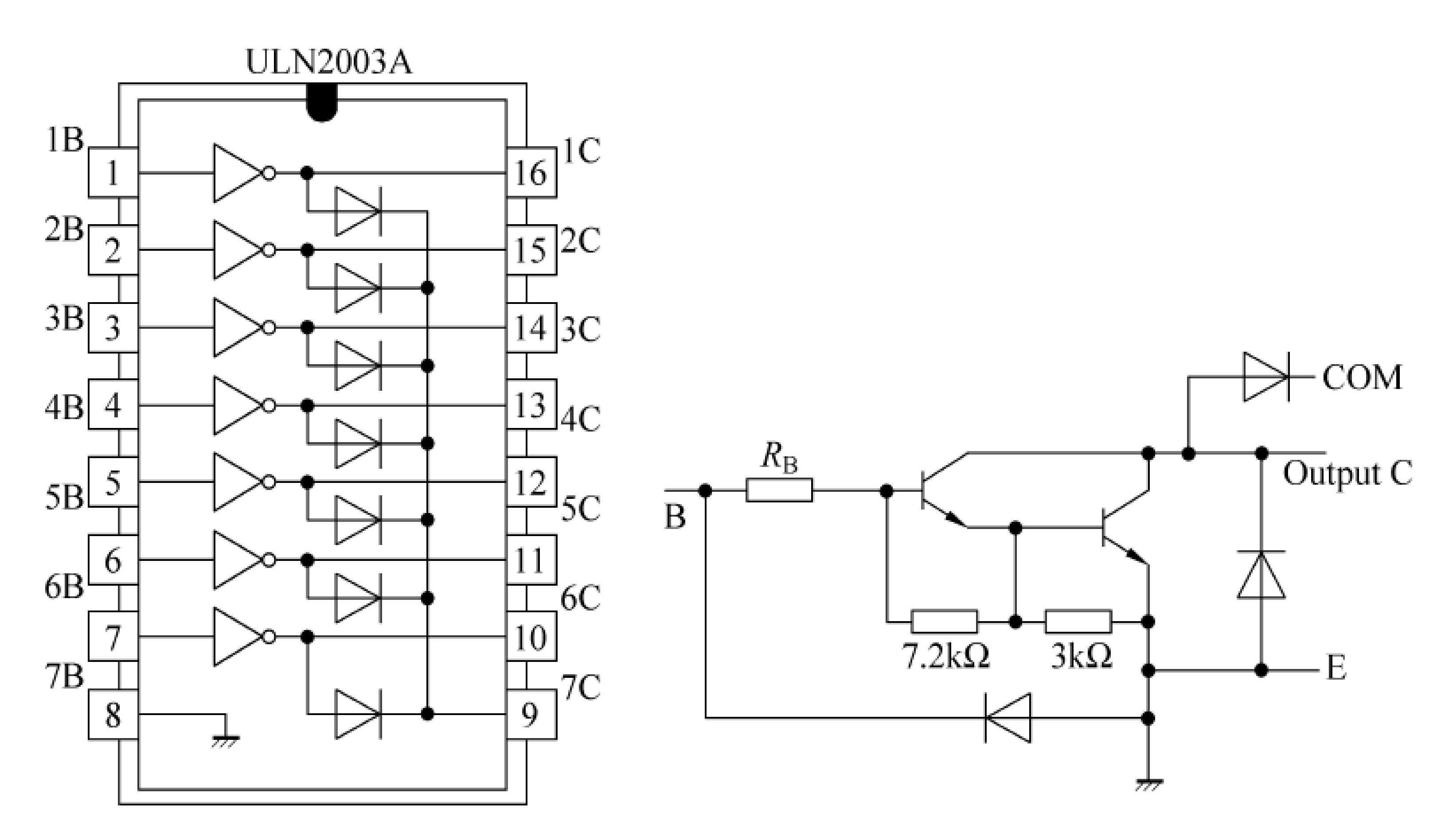


图 3.35　ULN2003A 达林顿阵列

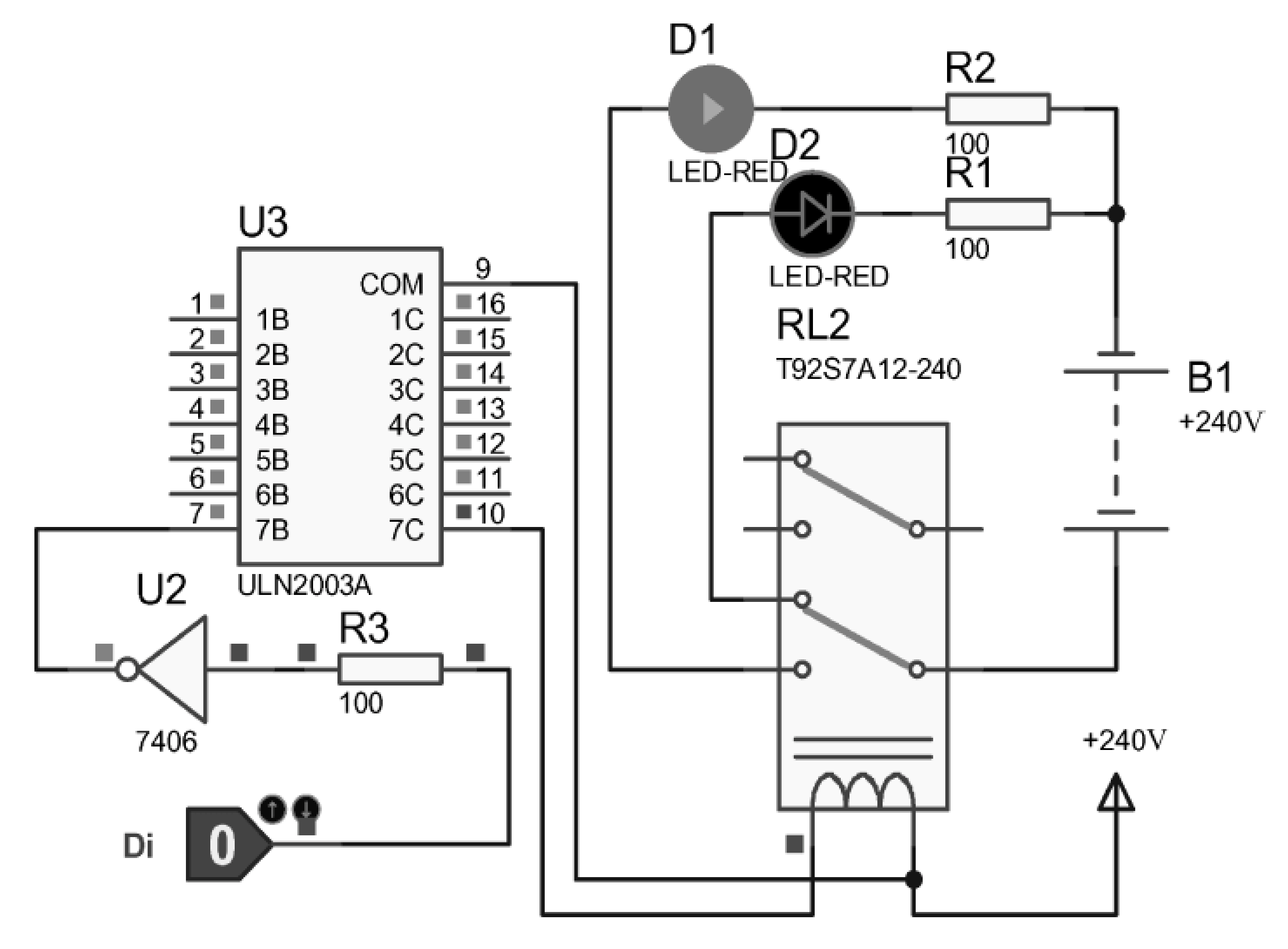


图 3.36　达林顿阵列输出驱动继电器

2）晶闸管驱动电路

晶闸管又称可控硅(Silicon Controlled Rectifier，SCR)，是可控硅整流元件的简称，是一种大功率的半导体器件，具有用小功率控制大功率、开关无触点等特点，在交直流电机调速系统、随动系统等工业控制领域应用广泛。晶闸管是一个三端器件，有单向晶闸管与双向晶闸管之分。单向晶闸管有阳极 A、阴极 K、控制极(门极)G 三个极。当阳、阴极之间加正压时，控制极与阴极两端施加正压使控制极电流增大到触发电流值时，晶闸管由截止转为导通；只有在阳、阴极间施加反向电压或阳极电流减小到维持电流以下，晶闸管由导通变为截止。单向晶闸管具有单向导电功能，在控制系统中多用于直流大电流场合，也可在交流系统中用于大功率整流回路。双向晶闸管也叫三端双向可控硅，在结构上相当于两个单向晶闸管的反向并联，但共享一个控制极，当两个电极之间的电压大于 1.5V 时，不论极性如何，便可利用控制极 G 触发电流控制其导通。双向晶闸管具有双向导通功能，因此特别适用于交流大电流场合。

晶闸管常用于高电压大电流的负载，因此不适宜与 CPU 直接相连，在实际使用时要采用隔离措施。图 3.37 为经光耦隔离的双向晶闸管驱动电路，当 CPU 数据线 D_i 输出“1”时，经 7406 反相变为低电平，光耦二极管导通，使光敏晶闸管导通，导通电流再触发双向晶闸管导通，从而驱动大型交流负荷设备 *RL*。图中电流方向为 Proteus 软件仿真时所显示瞬时方向。

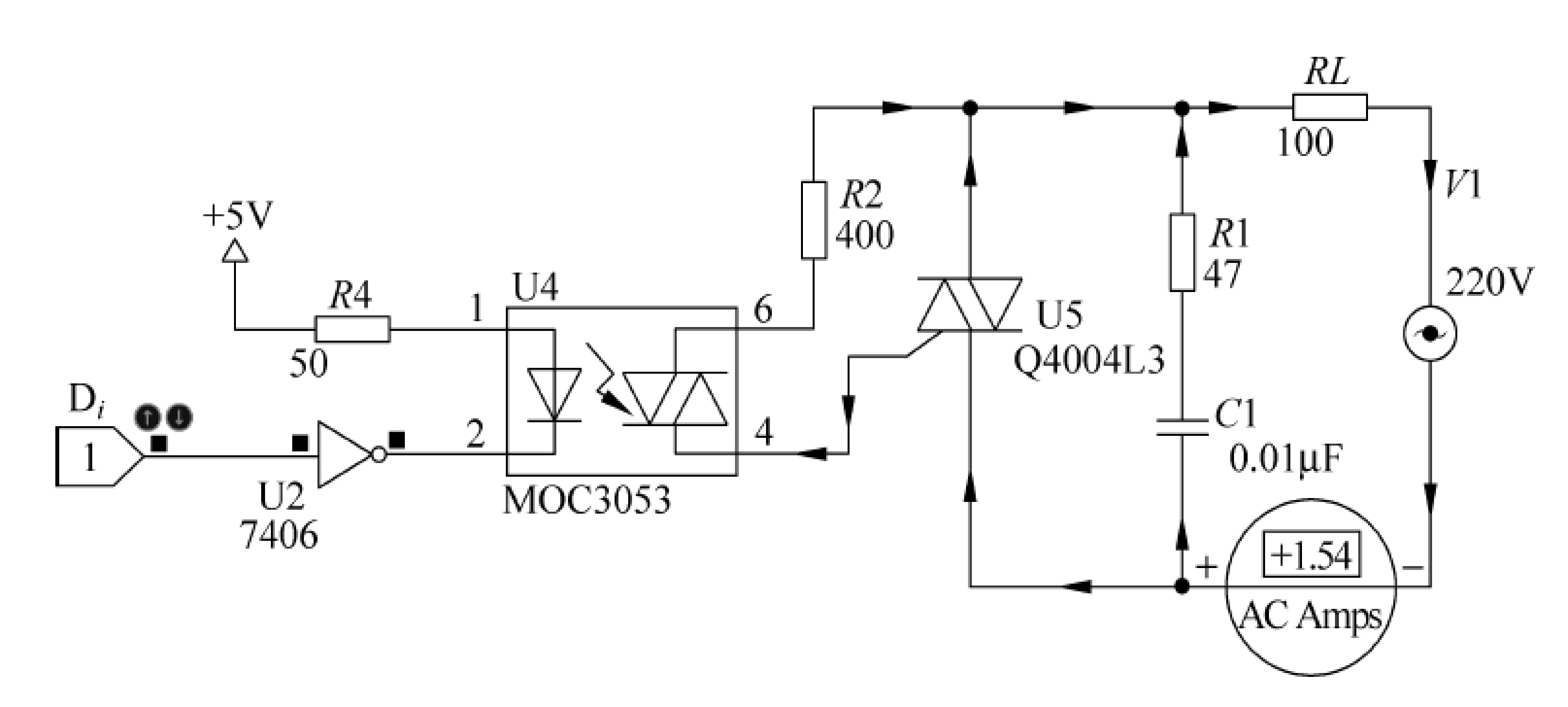


图 3.37　经光耦隔离的双向晶闸管驱动电路

3）固态继电器驱动电路

固态继电器(Solid State Relay,SSR)是由微电子电路、分立电子器件、电力电子功率器件组成的无触点开关，能实现电磁继电器的功能，故称为固态继电器。它用隔离器件实现控制端与负载端的隔离；其输入端用微小的控制信号，达到直接驱动大电流负载的作用。它具有体积小、开关速度快、无机械噪声、无抖动和回跳、寿命长等传统继电器无法比拟的优点，在计算机控制系统中得到广泛的应用。

固态继电器 SSR 是一个四端组件，有两个输入端、两个输出端，图 3.38 所示为其结构原理图，共由五部分组成：光电隔离电路的作用是在输入与输出之间起信号传递作用，同时使两端在电气上完全隔离；控制触发电路是为后级提供一个触发信号，使电子开关(三极管或晶闸管)能可靠地导通；电子开关电路用来接通或关断直流(或交流)负载电源；吸收保护电路的功能是为了防止电源的尖峰和浪涌对开关电路产生干扰造成开关的误动作或损害，一般由 *RC* 串联网络和压敏电阻组成；零压检测电路是为交流型 SSR 过零触发而设置

的。图 3.39 所示为固态继电器驱动电路。

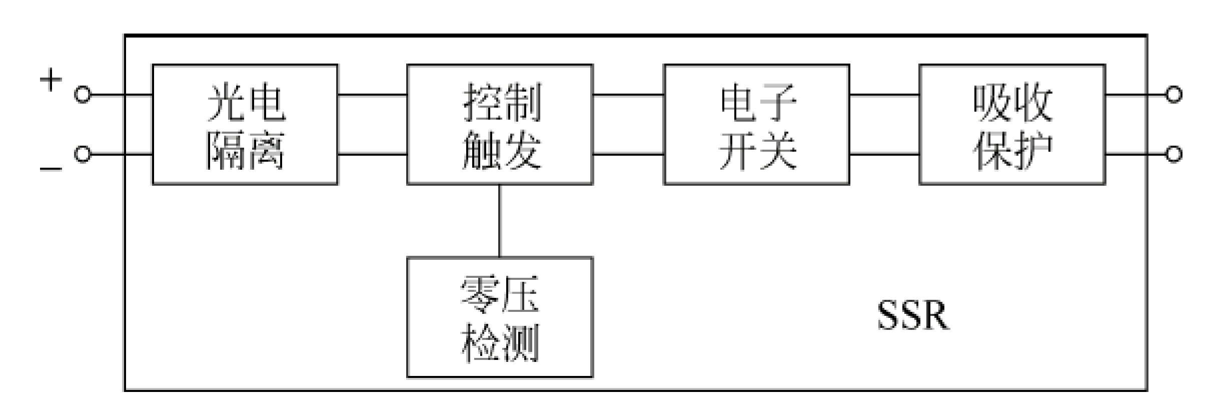


图 3.38 固态继电器结构原理图

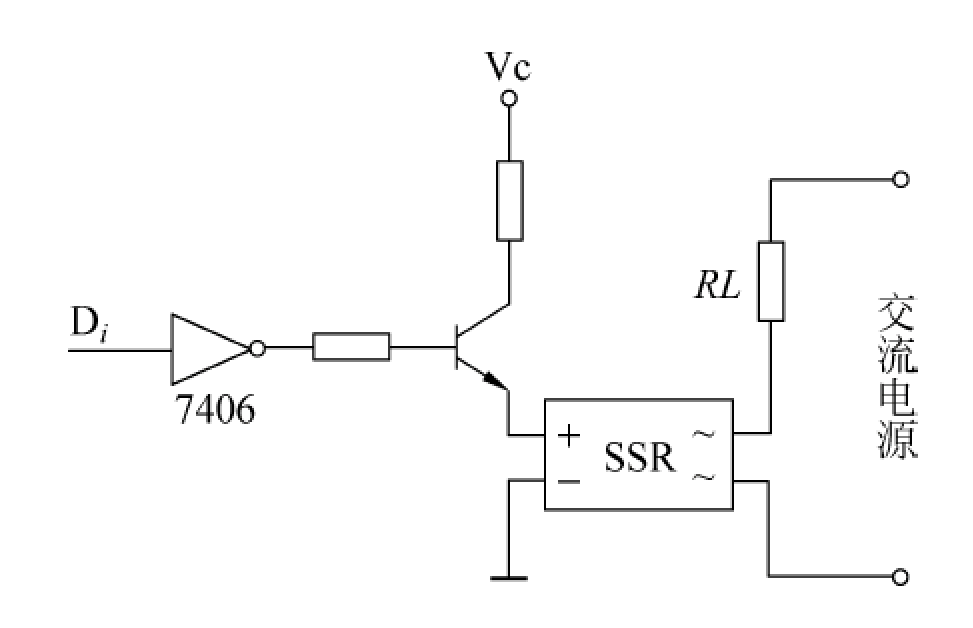


图 3.39 固态继电器驱动电路

3.4 模拟量输入输出通道

模拟量输入、输出通道是计算机与控制对象之间的重要接口,也是实现工业过程控制的重要组成部分。其主要功能是将生产过程中被测模拟量经传感器(或变送器)测量、信号调理、放大、采样保持,再经 A/D 转换器转换成数字量,并通过适当的接口送入控制计算机。在这里,把 A/D 转换器及其相关接口电路称为模拟量输入通道,又称为 A/D 通道或 AI 通道,如图 3.40 所示;同样,控制计算机处理后的数据需通过相应的接口和 D/A 转换器转换成模拟量,再经放大驱动电路输出,送执行机构完成一定的控制任务。在这里,把 D/A 转换器及相应的接口称为模拟量输出通道,又称为 D/A 通道或 AO 通道。

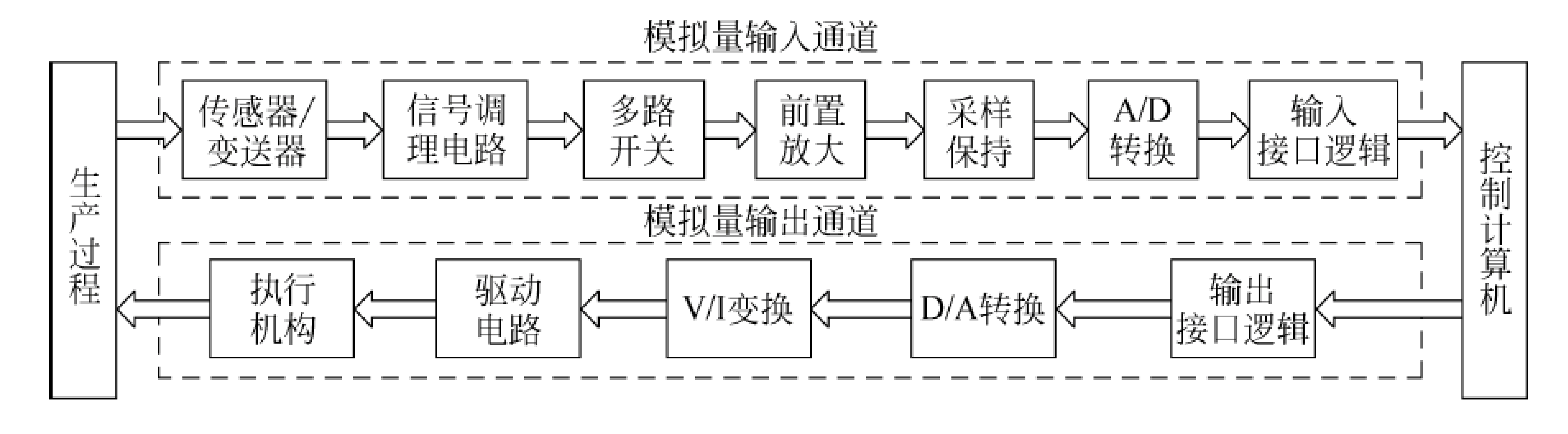


图 3.40 模拟量输入输出通道结构图

3.4.1 模拟量输入通道

模拟量输入通道主要将被控对象的过程参数如温度、压力、流量、液位、重量等模拟量信号转换成计算机可以接收的数字量信号,并经相应接口送计算机处理。主要包括传感器(或

变送器)、信号调理电路、多路选择开关、前置放大电路、采样保持器、A/D 转换器、输入接口等六部分。该通道的核心是模/数转换器即 A/D 转换器,如图 3.40 所示。

根据应用的不同,模拟量输入通道可以有不同的结构形式。其中传感器和变送器有关知识已在"自动化仪表"或"检测技术"相关课程中有所学习,这里不再重复;输入接口部分因为是模拟量经 A/D 转换之后的数字量接口,因此可参考数字量输入通道的地址译码电路及输入缓冲电路;下面主要针对信号调理、多路选择、前置放大、采样保持、A/D 转换几部分一一讨论。

1. 信号调理

模拟量输入通道的信号调理电路是传感器和 A/D 转换器之间的桥梁,主要完成信号的转换、保护、滤波、线性化、隔离等。

1) 信号的转换

信号的转换包括信号标度变换及信号形式的变换,标度变换是把经由各种传感器所得到的不同种类和不同电平的被测模拟信号变换成统一的标准信号,如:0～10mA 信号、4～20mA 信号或 0～±5V 信号,标度变换器一般包含在变送器中。信号形式的变换是指对被控量的检测往往采用各种类型的测量变送器,它们的输出信号常为 0～10mA 或 4～20mA 的电流信号,可采用电阻分压法把现场传送来的电流信号转换为电压信号,即 0～±5V 信号,以下是两种变换电路。

(1) 无源 *I*/*V* 变换

无源 *I*/*V* 变换电路是利用无源器件电阻来实现,再加上 *RC* 滤波和二极管限幅等保护,如图 3.41(a)所示,其中 *R*2 为精密电阻。对于 0～10mA 输入信号,可取 *R*1＝100Ω,*R*2＝500Ω,这样当输入电流在 0～10mA 变化时,输出的电压就为 0～5V 范围;对于 4～20mA 输入信号,可取 *R*1＝100Ω,*R*2＝250Ω,这样输出的电压为 1～5V。

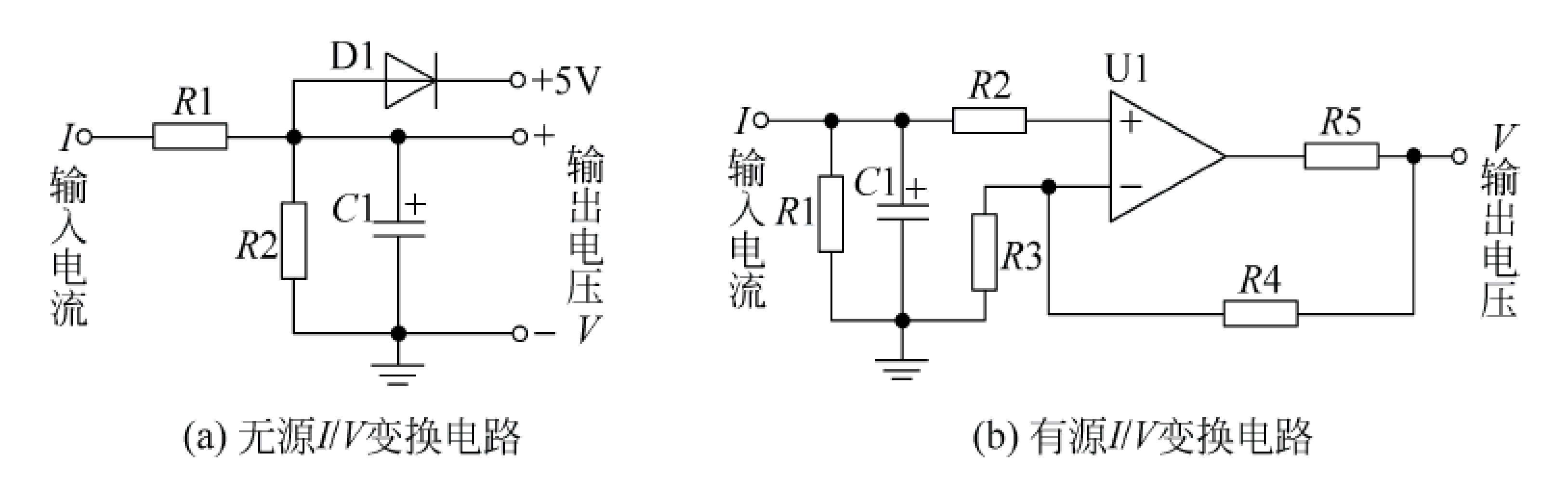


图 3.41 *I*/*V* 变换电路

(2) 有源 *I*/*V* 变换

有源 *I*/*V* 变换是利用有源器件运算放大器和电阻电容组成,如图 3.41(b)所示。利用同相放大电路,把电阻 *R*1 上的输入电压变成标准输出电压。该同相放大电路的放大倍数为 $G=V/(I\cdot R1)=1+R4/R3$。若取 *R*1＝200Ω,*R*3＝100kΩ,*R*4＝150kΩ,则输入电流 *I* 的0～10mA 就对应电压输出 *V* 的 0～5V;若取 *R*1＝200Ω,*R*3＝100kΩ,*R*4＝25kΩ,则 4～20mA 的输入电流对应于 1～5V 的电压输出。

(3) 专用 *I*/*V* 转换集成电路

RCV420 是一种精密的专用 *I*/*V* 转换器,它能将 4～20mA 的环路电流变换成 0～5V

的电压输出，是目前最佳的 I/V 转换电路方案，具有可靠的性能和较低的成本，有商用级（0～70℃）和工业级（－25～＋85℃）可供选择。广泛应用于过程控制、工业控制、数据采集和信号变换等领域。

RCV420 的内部电路主要由精密运算放大器、电阻网络和 10V 基准电源组成，其引脚功能见表 3.2，基本应用电路如图 3.42 所示。在一般使用时，10、11、12 引脚相连，14、15 引脚相连，5、2、13 引脚接地，7、8 引脚悬空，电源 16 引脚和 4 引脚与地之间接入 1μF 电容，输入信号是连接＋IN 还是－IN，取决于输入信号的极性。

表 3.2　RCV420 引脚功能

引　　脚	符　　号	功　　能
1	－IN	负输入端
2	C_T	输入中心抽头
3	＋IN	正输入端
4	$V-$	负电源
5	Ref Com	基准公共端
6	NC	空端
7	Ref Noise Reduction	基准降噪端
8	Ref Trim	基准调整端
9	NC	空端
10	Ref f_B	基准反馈端
11	Ref OUT	基准输出端
12	Ref IN	基准输入端
13	Rcv Com	接收公共端
14	Rcv OUT	接收输出端
15	Rcv f_B	接收反馈端
16	$V+$	正电源

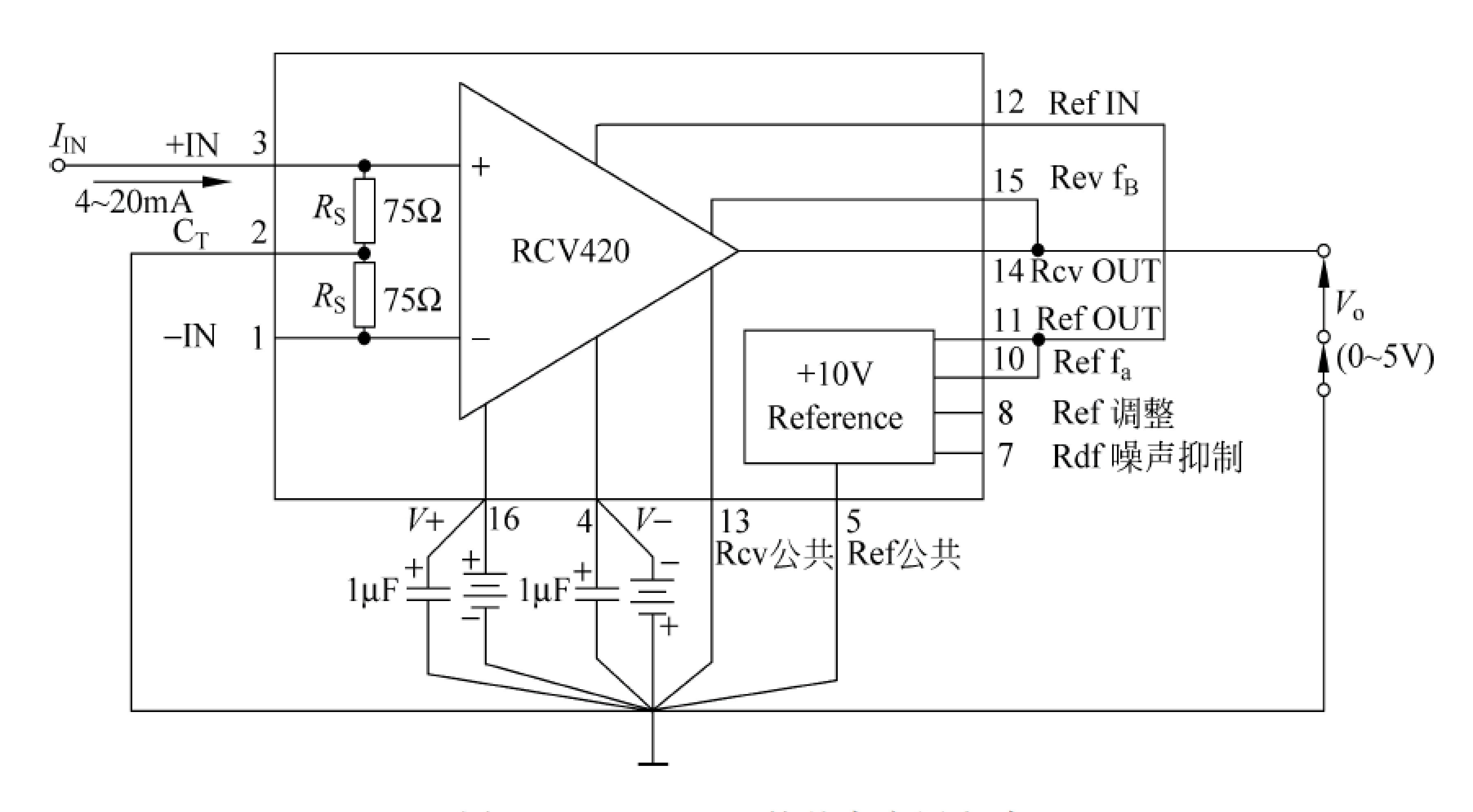


图 3.42　RCV420 的基本应用电路

RCV420 可与 XTR101/102/103（充当变送器）等器件结合使用，完成远距离的测温。即 XTR101 将热电偶信号（温度）变送成 4～20mA 的电流输出，经远距离传输后，由 I/V 变

换器 RCV420 将其变换成 0～5V 的电压信号，该信号可直接与 A/D 转换器及智能仪表连接构成远距离测温系统。

2）保护

当从生产现场取得传感器信号或其他模拟信号时，如果信号线漏电，或是出现浪涌噪声电压，以及前级设备出现故障，都可能在输入端出现过高电压而损坏设备，因此在输入端可加入保护电路，进行过电压保护、过电流保护、防静电保护等。

3）滤波

滤波电路主要用于滤掉或消除干扰信号，保留或增强有用信号。可采用有源滤波器或无源滤波器。

4）线性化处理

有些电信号转换后与被测参量呈现非线性。所以必须对信号进行线性化处理，使它接近线性化。在硬件上可采用加负反馈放大器或采用线性化处理电路；软件上可采用分段线性化数字处理的办法来解决。

5）模拟信号的隔离

数字信号的隔离用光电耦合器就可以了，模拟信号的隔离比较复杂，可以转换成数字信号后再采取光电隔离，并在供电时采用 DC/DC 隔离；也可以用隔离放大器如 BB 公司的 ISO 系列；还可以通过高频转换，用变压器隔离，如工业用温度变送器，信号隔离器等。

2. 多路选择

由于计算机的工作速度远远高于被测模拟量的变化，因此一台计算机系统可监控几十个外部变量，但在某一时刻计算机只能接收一个回路的信号。这就必须通过多路模拟开关实现多选一的操作，使多路输入信号依次地切换到后级。

计算机控制系统使用的多路开关种类很多，并具有不同的功能和用途。如集成电路芯片 CD4051（双向、单端、8 路）、CD4052（单向、双端、4 路）、AD7506（单向、单端、16 路）等。所谓双向，就是该芯片既可以实现多到一的切换，也可以完成一到多的切换；而单向则只能完成多到一的切换。双端是指芯片内的一对开关同时动作，从而完成差动输入信号的切换，以满足抑制共模干扰的需要。

下面以常用的 CD4051 为例，其结构原理如图 3.43 所示。CD4051 由电平转换、译码驱动及开关电路三部分组成。当禁止端 $\overline{INH}$ 为“1”时，前后级通道断开，即 S_0～S_7 端与 S_m 端不可能接通；当为“0”时，则通道可以被接通，通过改变控制输入端 C、B、A 的数值，就可选通 8 个通道 S_0～S_7 中的一路。例如，当 $CBA=000$ 时，通道 S_0 选通；当 $CBA=001$ 时，通道 S_1 通；当 $CBA=111$ 时，通道 S_7 选通。

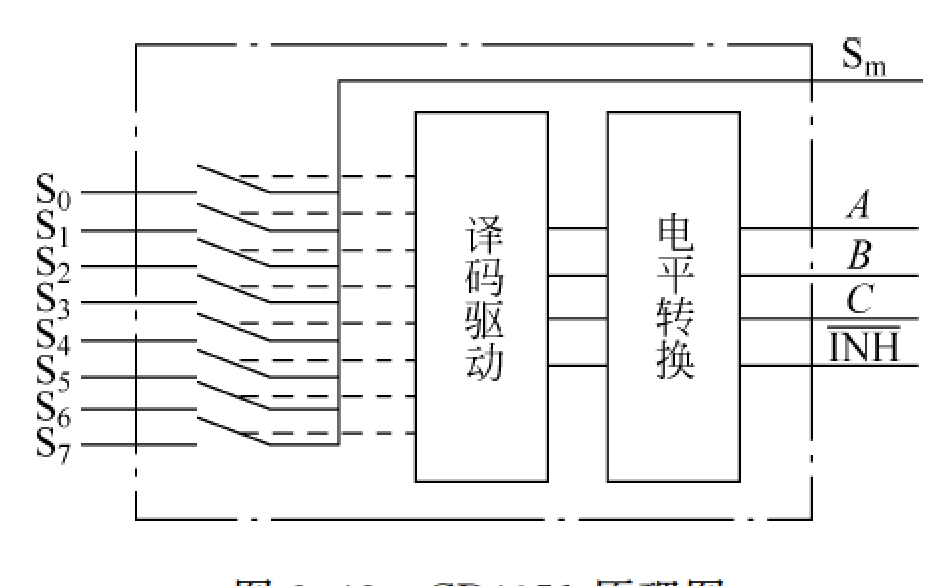


图 3.43 CD4051 原理图

当采样通道多至 16 路时，可直接选用 16 路模拟开关的芯片，也可以将两个 8 路 4051 并联起来，组成一个单端的 16 路开关。图 3.44 给出了两个 CD4051 扩展为 1×16 路模拟开关的电路。数据总线 D_3～D_0 作为通道选择信号，D_3 用来控制两个多路开关的禁止端。

当 $D_3=0$ 时，选中上面的多路开关，此时当 D_2、D_1、D_0 从 000 依次变为 001、010、…、111，则依次选通 $S_0\sim S_7$ 通道；当 $D_3=1$ 时，选中下面的多路开关，此时当 D_2、D_1、D_0 依次从 000 变至 111，则依次选通 $S_8\sim S_{15}$ 通道。从而组成一个 16 路的模拟开关。

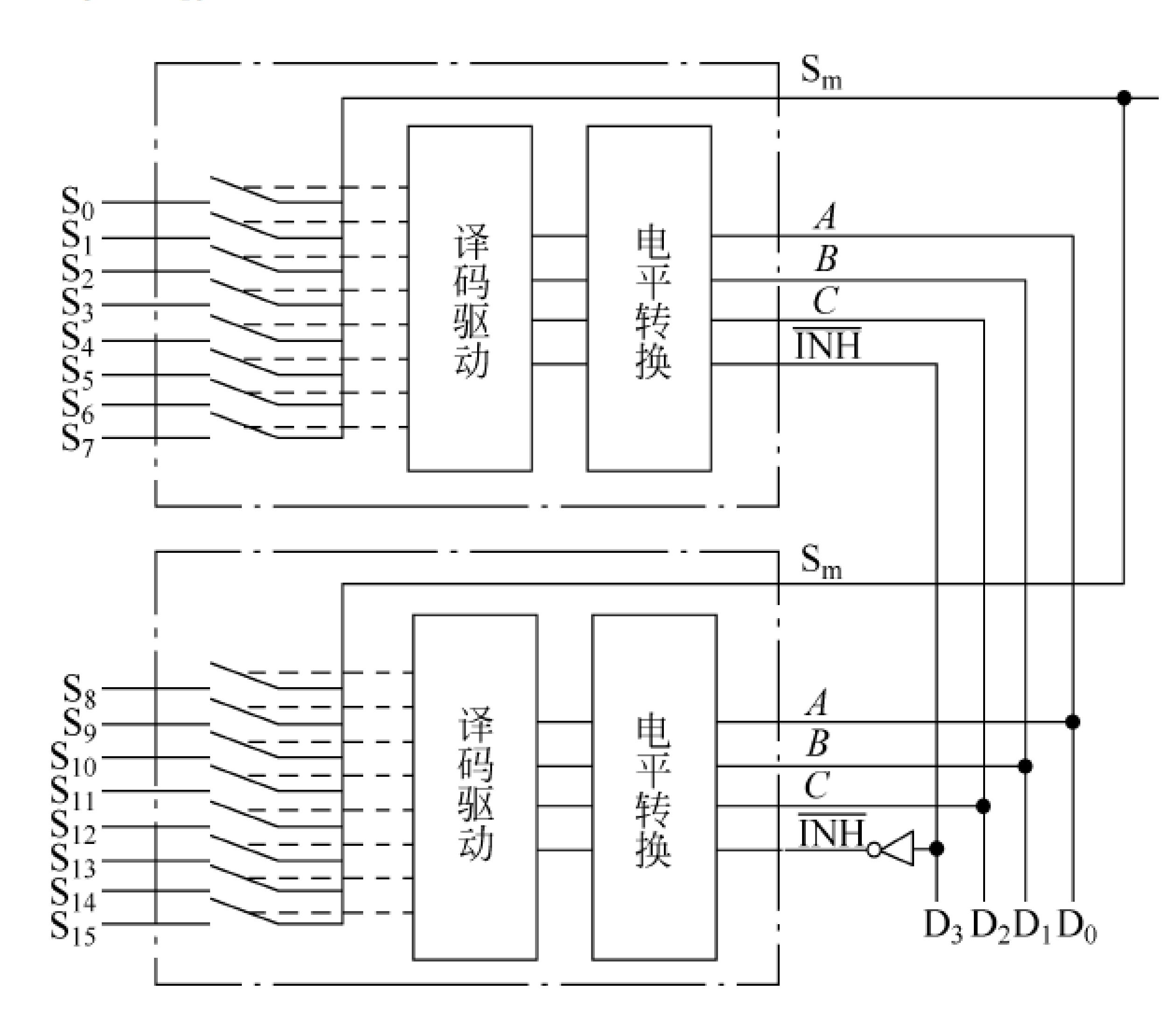


图 3.44　16 路模拟开关电路

3. 前置放大

前置放大是将输入的模拟小信号放大到 A/D 转换的量程范围之内，如 0～5V DC。对单纯的微弱信号，可用一个运算放大器进行单端同相放大或单端反相放大。而在实际工程中，来自生产现场的传感器信号往往带有较大的共模干扰，单个运放电路的差动输入难以起到很好的抑制作用，因此 A/D 通道中的前置放大器常采用由一组运放构成的测量放大器，也称仪表放大器，如图 3.45 所示。此测量放大器是由三个运放组成的对称结构，测量放大器的差动输入端 V_{IN+} 和 V_{IN-} 分别是两个运放 A2、A1 的同相输入端，输入阻抗很高，而且完全对称地直接与被测信号相连，因而有着极强的抑制共模干扰能力；R_G 是外接电阻，$R_2=R_4$，$R_1=R_3$，$R_5=R_6$ 专用来调整放大器的增益，增益公式为：

$$G=\frac{V_{OUT}}{V_{IN+}-V_{IN-}}=\frac{R_2}{R_1}\left(1+\frac{2R_5}{R_G}\right) \tag{3.1}$$

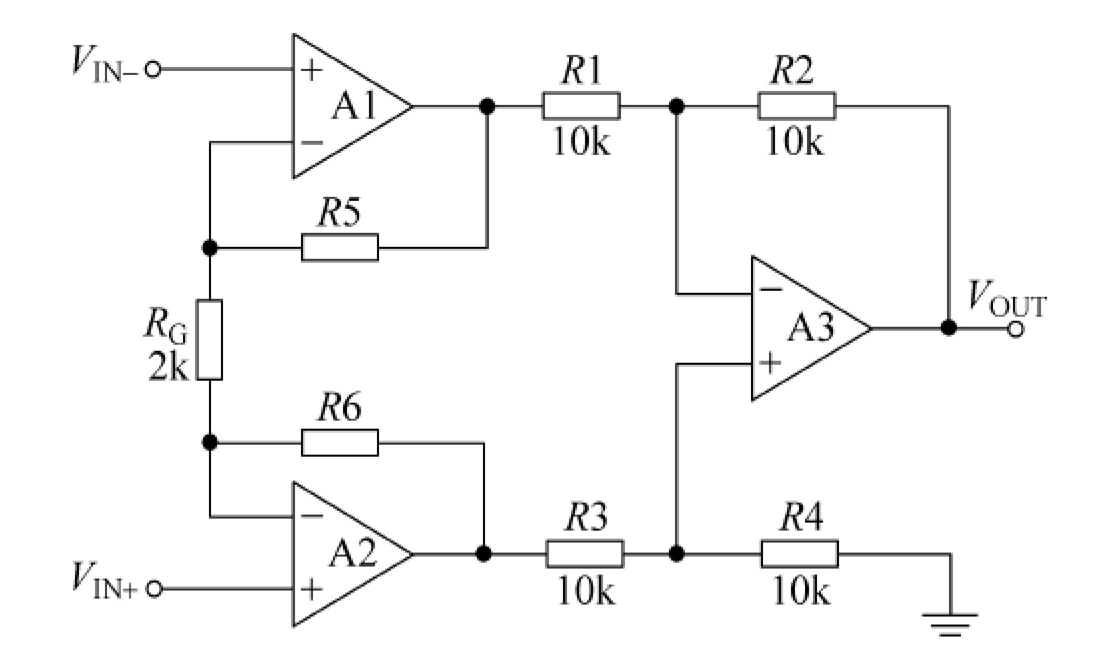


图 3.45　经典测量放大器电路

这种测量放大器的集成电路芯片有多种，如 AD521/522、INA102 等。

在 A/D 转换通道中，多路被测信号常常共用一个测量放大器，而各路的输入信号大小往往不同，却都要放大到 A/D 转换器的同一量程范围，因此，测量放大器的增益也需要不同，具有这种性能的放大器称为可变增益放大器或可编程增益放大器(Programmable Gain Amplifier，PGA)。它是一种通用性很强的放大器，其放大倍数可以根据需要用程序进行控制，采用这种放大器，可使 A/D 转换器满量程信号达到均一化，因而大大提高测量精度。可编程增益放大器有两种：组合 PGA 和集成 PGA。

组合 PGA 一般由运算放大器、仪器放大器或隔离型放大器再加上一些其他附加电路组成。其工作原理是通过程序调整多路转换开关接通的反馈电阻的数值，从而调整放大器的放大倍数。

专门设计的可编程增益放大器电路即集成 PGA。集成 PGA 电路的种类很多，如美国微芯(Microchip)公司生产的 MCP6S21、MCP6S22、MCP6S26、MCP6S28 系列，美国模拟仪器公司(Analog Devices)生产的 AD8321、AD526 等，都属于可编程增益放大器。下面以 AD526 为例说明这种电路的原理及应用。

可编程放大器 AD526 是可通过软件对增益进行编程的单端输入的仪用放大器，器件本身所提供的增益是×1、×2、×4、×8、×16 五挡。它是一个完整的包括放大器、电阻网络和 TTL 数字逻辑电路的器件，使用时不需外加任何元件就可工作。其典型应用电路如图 3.46 和图 3.47 所示。AD526 各引脚功能如表 3.3 所示。

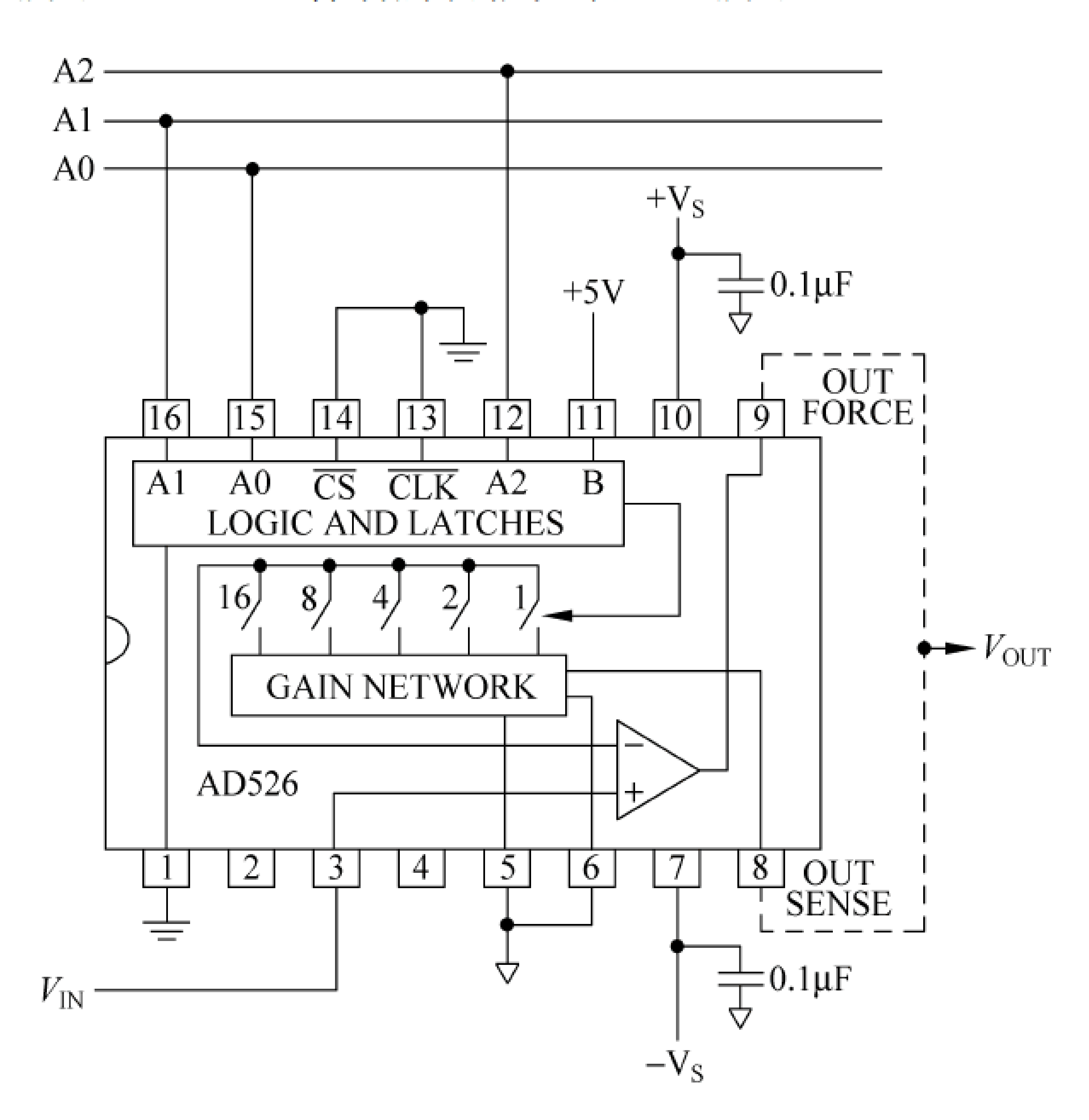


图 3.46　AD526 的透明工作模式

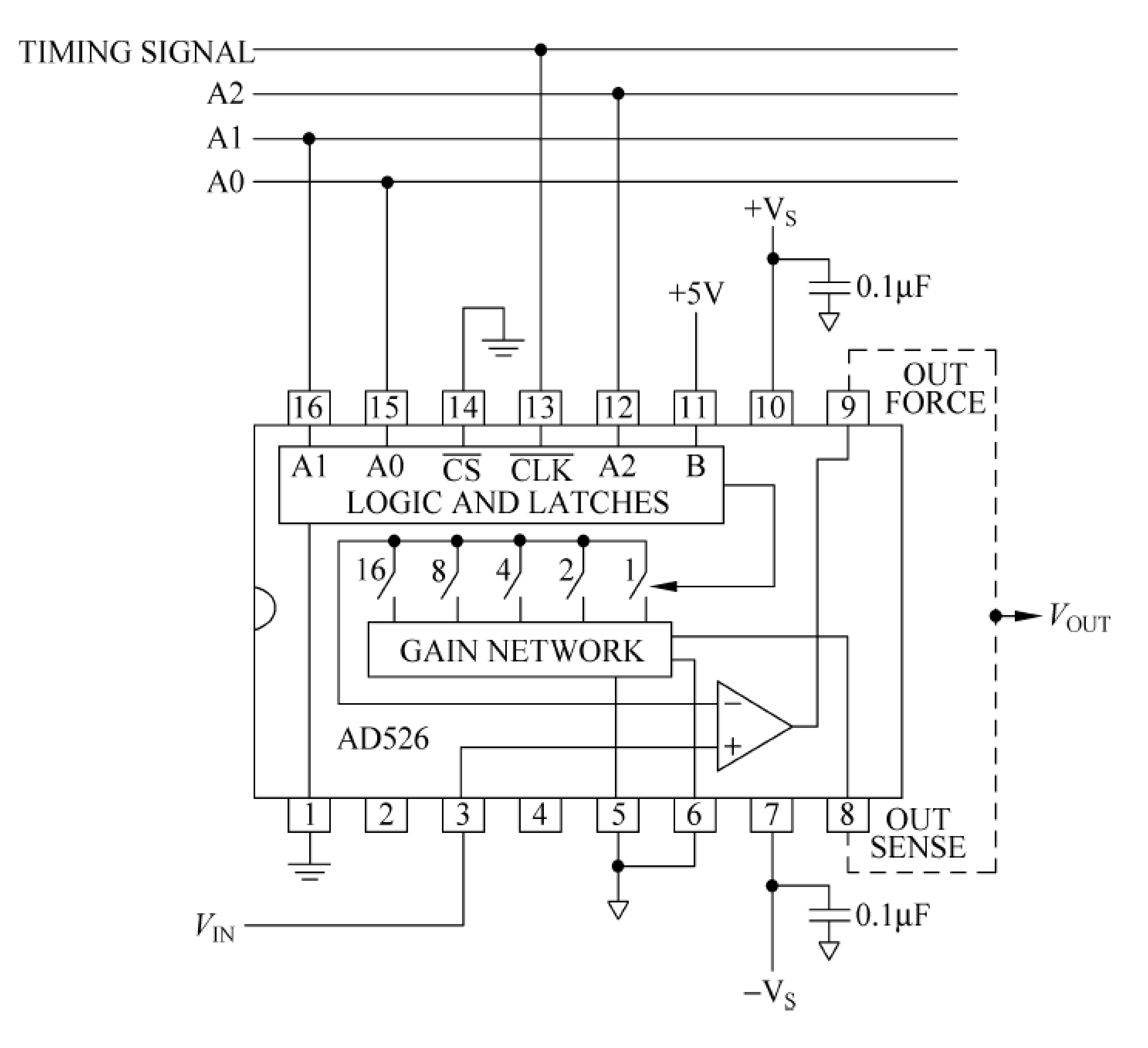


图 3.47　AD526 的锁存工作模式

表 3.3　AD526 引脚功能

引　　脚	符　　号	功　　能
1	DIG GND	数字地
2、4	NULL	输入失调电压调零端，引脚 2 和 4 通过电位器相连
3	V_{IN}	信号输入端
5	ANALOG GND2	模拟地 2
6	ANALOG GND1	模拟地 1，该引脚与 5 相连后接地
7	$-V_S$	负电源输入端，典型值为－15V
8	V_{OUT} SENSE	输出检测端
9	V_{OUT} FORCE	输出端，该引脚与 8 相连后作为输出
10	$+V_S$	正电源输入端，典型值为＋15V
13	$\overline{\text{CLK}}$	时钟输入端，$\overline{\text{CLK}}$＝0 为透明工作模式，$\overline{\text{CLK}}$＝1 为锁存工作模式，不用时应接地
14	$\overline{\text{CS}}$	片选端，低电平有效，不用时应接地
11、15、16、12	B、A0、A1、A2	增益控制码输入端

由图 3.46 和图 3.47 可见，AD526 有两种工作模式，$\overline{\text{CLK}}$＝0 为透明工作模式：当 B 为"1"，$\overline{\text{CS}}$和$\overline{\text{CLK}}$为"0"，输入模拟信号在 A2、A1、A0 到来时，增益立即响应，但不被保存；$\overline{\text{CLK}}$＝1 为锁存工作模式：当$\overline{\text{CLK}}$变为"1"时，增益码被锁存到内部寄存器中，直到$\overline{\text{CLK}}$从"1"回到"0"。增益控制状态表见表 3.4 所示。

表 3.4 AD526 增益控制状态表

增益码				控制	状态	
A2(12)	A1(16)	A0(15)	B(11)	$\overline{CLK}$($\overline{CS}$=0)	增益	模式
×	×	×	×	1	前状态	锁存
0	0	0	1	0	1	透明
0	0	1	1	0	2	透明
0	1	0	1	0	4	透明
0	1	1	1	0	8	透明
1	×	×	1	0	16	透明
×	×	×	0	0	1	透明
×	×	×	0	1	1	锁存
0	0	0	1	1	1	锁存
0	0	1	1	1	2	锁存
0	1	0	1	1	4	锁存
0	1	1	1	1	8	锁存
1	×	×	1	1	16	锁存

4. 采样保持

当某一通道进行 A/D 转换时，由于 A/D 转换需要一定的时间，如果输入信号变化较快，就会引起较大的转换误差。为了保证 A/D 转换的精度，需要应用采样保持电路。采样保持电路(Sample Hold Devices，S/H)，它的作用是采集模拟输入电压在某一时刻的瞬时值，并在模数转换器进行转换期间保持不变，以供模数转换。但如果输入信号变化很慢，如温度信号；或者 A/D 转换时间较短，使得在 A/D 转换期间输入信号变化很小，并在允许的 A/D 转换精度内，就不必再选用采样保持器。

1) 采样的基本原理

系统的采样形式有多种，包括周期采样、多阶采样和随机采样，应用最多的是周期采样。所谓周期采样就是以相同的时间间隔进行采样，如图 3.48 所示，即把一个连续变化的模拟信号 $y(t)$按一定的时间间隔 T 转变为在瞬时 0、T、$2T$、…、kT 的一连串脉冲序列信号$y^*(t)$(离散模拟信号)，其中间隔时间 T 称为采样周期，脉冲宽度 τ 称为采样宽度(或采样时间)。在实际系统中 $\tau \ll T$，也就是说，可以近似地认为采样信号 $y^*(t)$是 $y(t)$在采样开关闭合时的瞬时值。

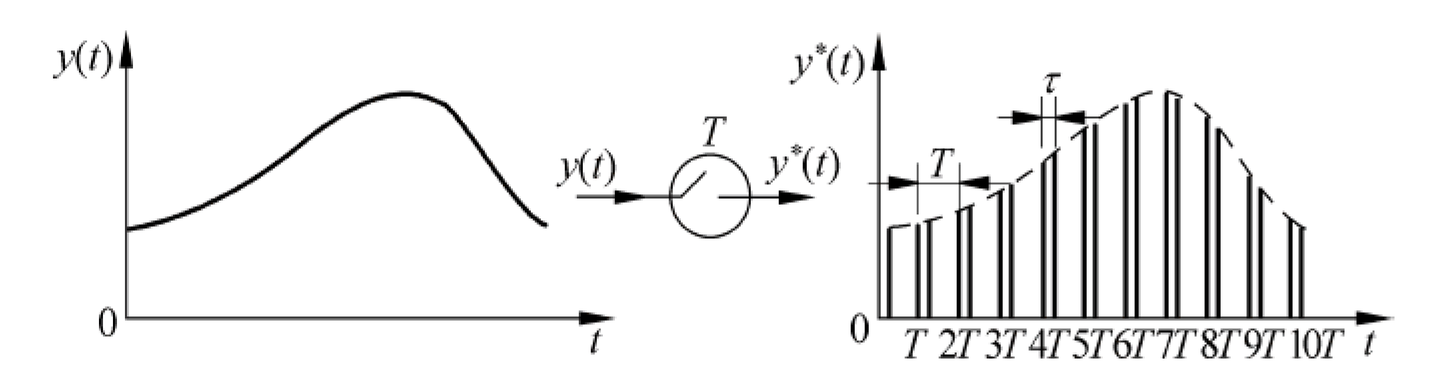


图 3.48 连续模拟信号的采样

采样过程可用数学描述为以下表达式(3.2)：

$$y^*(t) = y(t)\delta_T(t) = y(t)\sum_{k=-\infty}^{+\infty}\delta(t-kT) \tag{3.2}$$

其中 $\delta_T(t)=\sum_{k=-\infty}^{+\infty}\delta(t-kT)$ 为单位脉冲序列，$\delta(t)$ 的工程表达式为：

$$\delta(t)=\begin{cases}1 & t=0\\ 0 & t\neq 0\end{cases} \tag{3.3}$$

为了考虑工程上可实现性，式(3.2)又可近似为式(3.4)：

$$y^*(t)=y(t)\delta_T(t)=y(t)\sum_{k=0}^{+\infty}\delta(t-kT) \tag{3.4}$$

2）采样频率/周期的选取

由经验可知，采样频率(即采样周期的倒数，$f=1/T$)越高，采样信号 $y^*(t)$越接近原信号 $y(t)$，但若采样频率过高，在实时控制系统中将会把许多宝贵的时间用在采样上，从而失去了实时控制的机会；为了既不使采样信号 $y^*(t)$失真，又不会因频率太高而浪费时间，我们可依据香农采样定理进行采样。

香农定理指出：为了使采样信号 $y^*(t)$能完全复现原信号 $y(t)$，采样频率 f 至少要为原信号最高有效频率 f_{max}的 2 倍，即 $f\geqslant 2f_{max}$。采样定理给出了 $y^*(t)$唯一地复现 $y(t)$所必需的最低采样频率。但在实际应用中，常取 $f\geqslant(5\sim10)f_{max}$。

另外采样周期的选取还需遵循以下的一般原则。

(1) 系统受扰动情况：若扰动和噪声都较小，采样周期 T 应选大些；对于扰动频繁和噪声大的系统，采样周期 T 应选小些。

(2) 被控系统动态特性：滞后时间大的系统，采样周期 T 应选大些；对于快速系统，采样周期 T 应选小些。

(3) 控制品质指标要求：若超调量为主要指标，采样周期 T 应选大些；若希望过渡过程时间短些，采样周期 T 应选小些。

(4) 控制回路的数量：控制回路较多的系统，要考虑到能够处理完各回路所需时间来确定采样周期。

(5) 系统控制算法的类型：例如 PID 算法中微分作用和积分作用与采样周期相关，需综合考虑。

3）采样保持器

采样保持器(Sample and Hold Amplifier，S/H)是一种在 A/D 转换时可以保持住输入模拟信号的电平，并在 A/D 转换后跟踪输入信号变化的器件。它有两种工作模式：一种是采样模式，此时其输出跟踪模拟输入电压；一种是保持模式，此时输出保持在命令发出时刻的输入模拟值上直至保持命令撤销。其主要由保持电容 CH、输入输出缓冲器(A1、A2)、模拟开关 S 及控制电路组成，如图 3.49 所示，内部输入输出缓冲器采用串联型接法。在采样时刻，加到模拟开关 S 上的信号使之接通，使存储元件(保持电容 CH)两端的电压随被采样信号变化；当采样终止时，模拟开关上的信号使之断开，输出值则保持在断开瞬间的值 V_c 不变。这里，缓冲放大器的作用是可以放大采样信号，同时提高采样保持器的输入阻抗、减小输出阻抗，以便与信号源和负载连接。它在电路中的连接方式有两种基本类型：一种是将信号先放大再存储，另一种是先存储再放大。

随着大规模集成电路的发展，目前有各种各样的采样/保持器集成电路。如用于一般目的的有 AD582、AD583、LF198/398 等；用于高速的有 THS-0025、THS-0060、THC-0030、

THC-1500 等；用于高分辨率的有 SHA1144、ADC1130 等。下面以 LF398 为例介绍采样保持的应用。

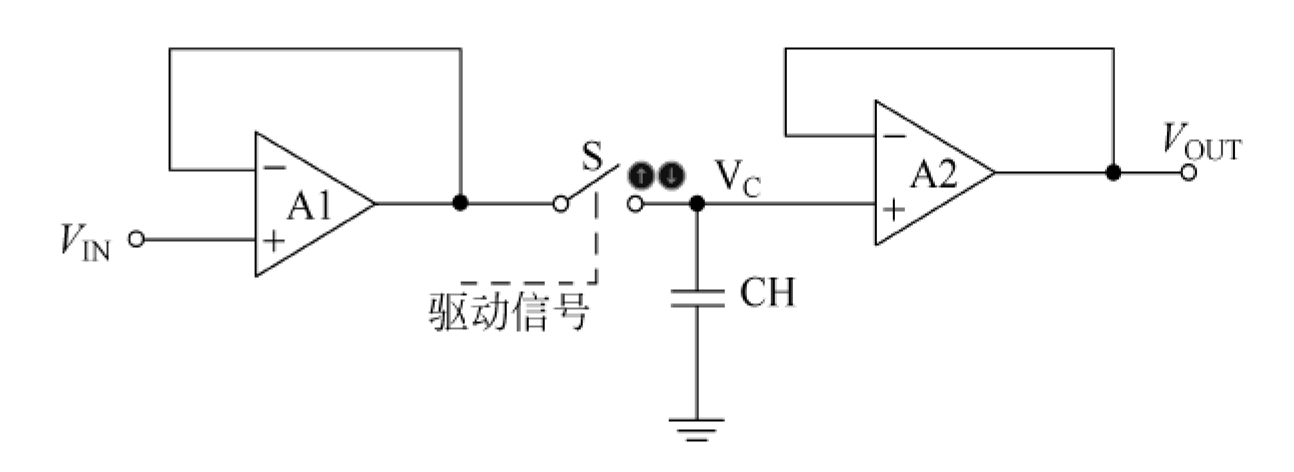
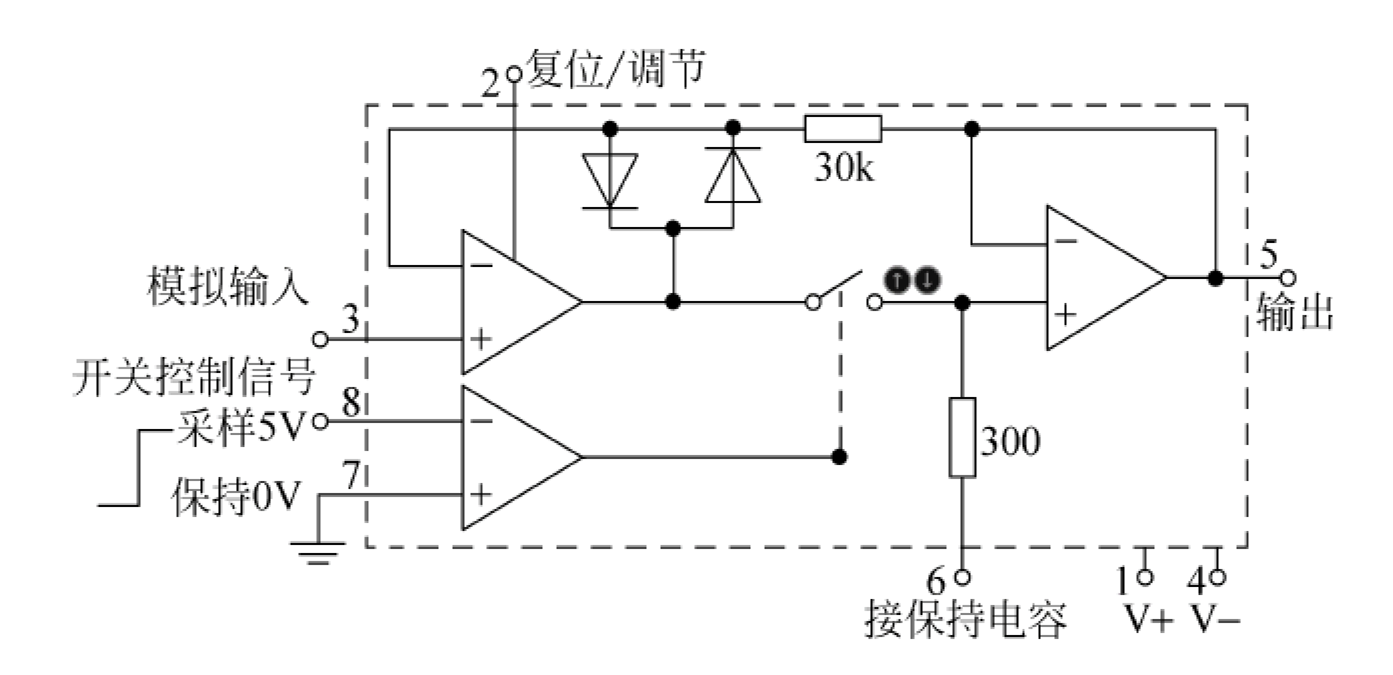


图 3.49　采样保持器的基本构成

LF398 的内部结构、引脚说明及典型应用电路如图 3.50 所示。内部输入输出放大器采用了反馈型的接法，这种接法采样精度更高(输出端只引入了输出缓冲器的失调电压，串联型则同时引入了输入、输出两个缓冲器的失调电压)；开关控制信号输入端可以直接与计算机数据口相连，LF398 的采样控制电平为“1”，保持电平为“0”，与 AD582 相反；操作电源为 ±5～±18V；6 引脚接保持电容，电容一般在 1000pF～1μF 之间，可根据实际需要选取；3 引脚接输入模拟信号，可连接多路选择器的输出；5 引脚接输出信号，直接与 A/D 转换器输入端相连。



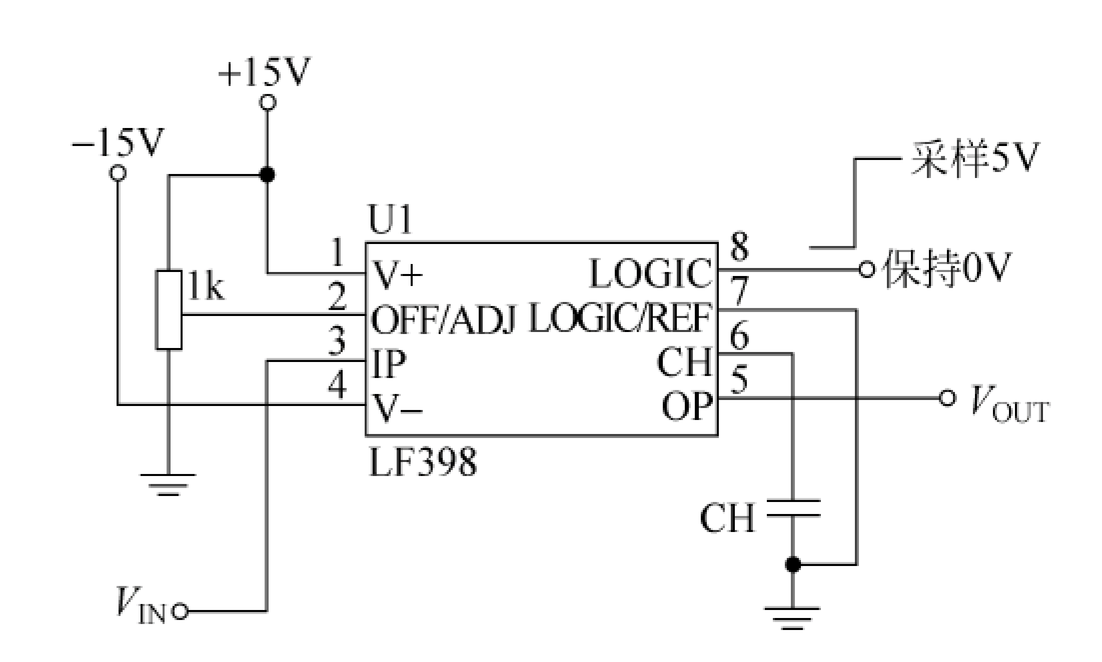


图 3.50　LF398 的内部结构及典型应用电路

5. A/D 转换

A/D 转换指将模拟信号转变为数字信号，A/D 转换器(Analog to Digital Converter，ADC)是将输入模拟信号转换成数字信号的装置。

1）A/D 转换的基本原理

A/D 转换常采用以下几种方式：计数比较型、双斜率积分型、逐次比较型、并行/串并行比较型、Σ-Δ 调制型、电容阵列逐次比较型、压频变换型等。

(1) 计数比较型

计数比较型主要由计数器、D/A 转换器(Digital to Analog Converter，DAC)及比较器组成。转换原理是：由计数器对固定频率信号 CLK 进行计数，计数输出值送 DAC，DAC 的输出模拟量 V_O 与输入模拟量 V_i 在比较器中进行比较，随着计数的进行，V_O 不断增加，当 $V_O \geqslant V_i$，计数器停止计数，此时的计数值即是模拟量 V_i 对应的数字量；这种 A/D 转换器结构简单，价格便宜，但转换速度比较慢，目前较少使用。

(2) 双斜率积分型

双斜率积分型 A/D 转换器转换原理如图 3.51 所示，在转换开始信号控制下，开关接通模拟量输入端，输入的模拟电压 V_{IN} 在固定时间 T 内对积分器上的电容 C 充电(正向积分)，时间一到，控制逻辑将开关切换到与 V_{IN} 极性相反的基准电源上，此时电容 C 开始放电(反向积分)，同时计数器开始计数。当比较器判定电容 C 放电完毕时就输出信号，由控制逻辑停止计数器的计数，并发出转换结束信号。这时计数器所记的脉冲个数正比于放电时间。放电时间 $T1$ 或 $T2$(见图 3.51(b))正比于输入电压 V_{IN}，即输入电压越大放电时间越长，计数器的计数值越大。因此，计数器计数值的大小反映了输入电压 V_{IN} 在固定积分时间 T 内的平均值。此种 A/D 转换器常见的有输出为 3 位半 BCD 码(二进制编码的十进制数)的 ICL7107、MC14433、输出为 4 位半 BCD 码的 ICL7135 等。其精度高，结构复杂，适用精度要求较高的场合。

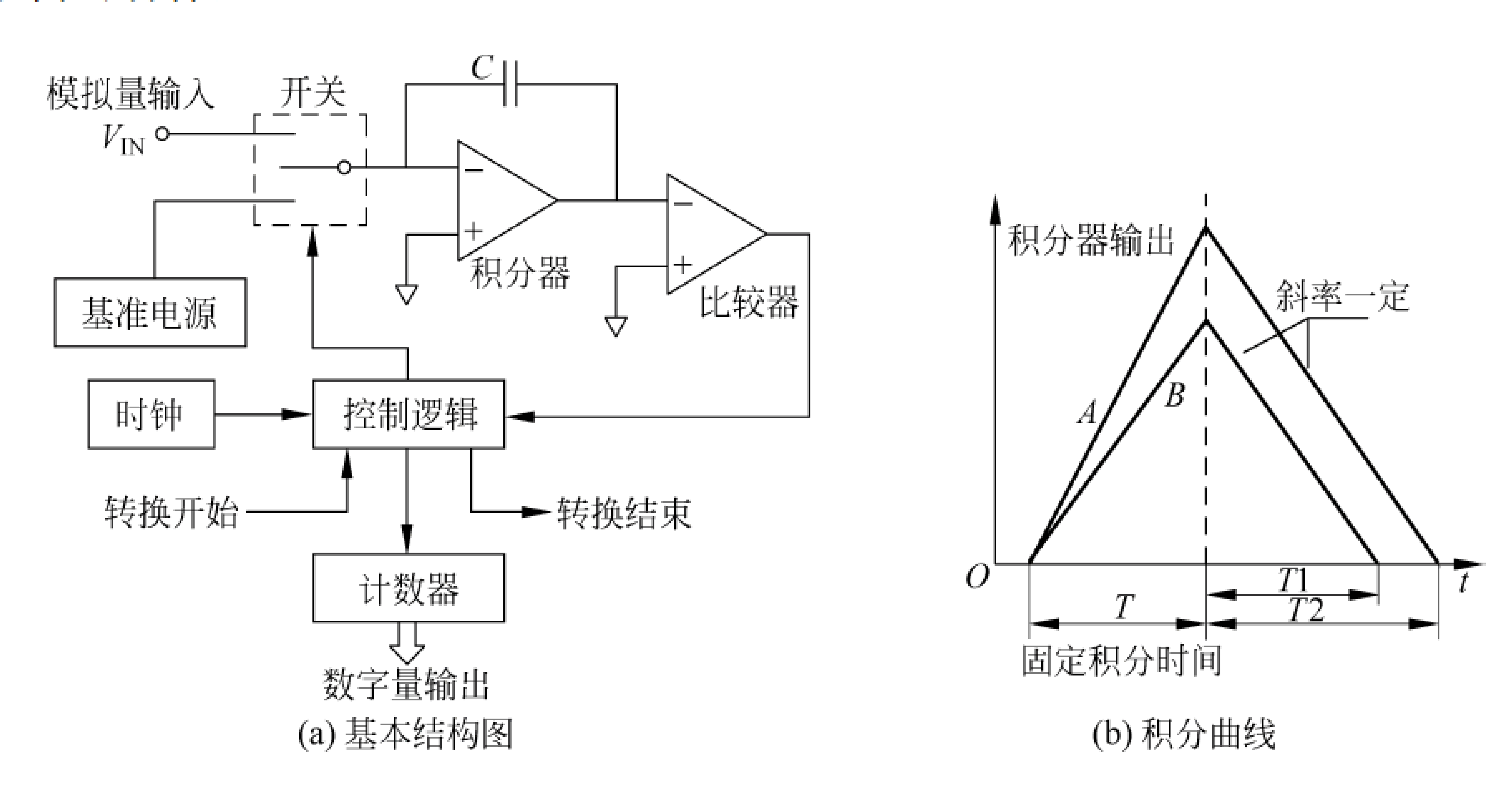


图 3.51 双斜率积分型 A/D 转换原理图

(3) 逐次比较型

逐次比较型 A/D 转换器(n 位)是由 n 位逐次逼近寄存器、n 位 D/A 转换器、比较器、逻辑控制电路、输出缓冲器等 5 部分组成，如图 3.52 所示。

下面以 4 位逐次比较型 A/D 转换器把模拟量 9 转换为二进制数 1001 为例，说明其 A/D 转换的工作原理：当转换开始信号作用后，时钟信号在控制逻辑作用下，首先使逐次逼

近寄存器的最高位 D3=1，其余为 0，此数字量 1000 经 D/A 转换器转换成模拟电压即 $V_O=8$，送到比较器输入端与被转换的模拟量 $V_{IN}=9$ 进行比较，控制逻辑根据比较器的输出进行判断。当 $V_{IN}\geqslant V_O$，则保留 D3 = 1；再对下一位 D2 进行比较，同样先使 D2=1，与 D3 一起即 1100 进入 D/A 转换器，转换为 $V_O=12$ 再进入比较器，与 $V_{IN}=9$ 比较，因 $V_{IN}<V_O$，则使 D2=0；再下一位 D1 位也是如此，D1=1 即 1010，经 D/A 转换为 $V_O=10$，再与 $V_{IN}=9$ 比较，因 $V_{IN}<V_O$，则使 D1=0；使最后一位 D0=1，即 1001 经 D/A 转换为 $V_O=9$，再与 $V_{IN}=9$ 比较，因 $V_{IN}\geqslant V_O$，保留 D0=1。比较完毕，寄存器中的数字量 1001 即为模拟量 9 的转换结果，存于输出锁存器中等待被读取。

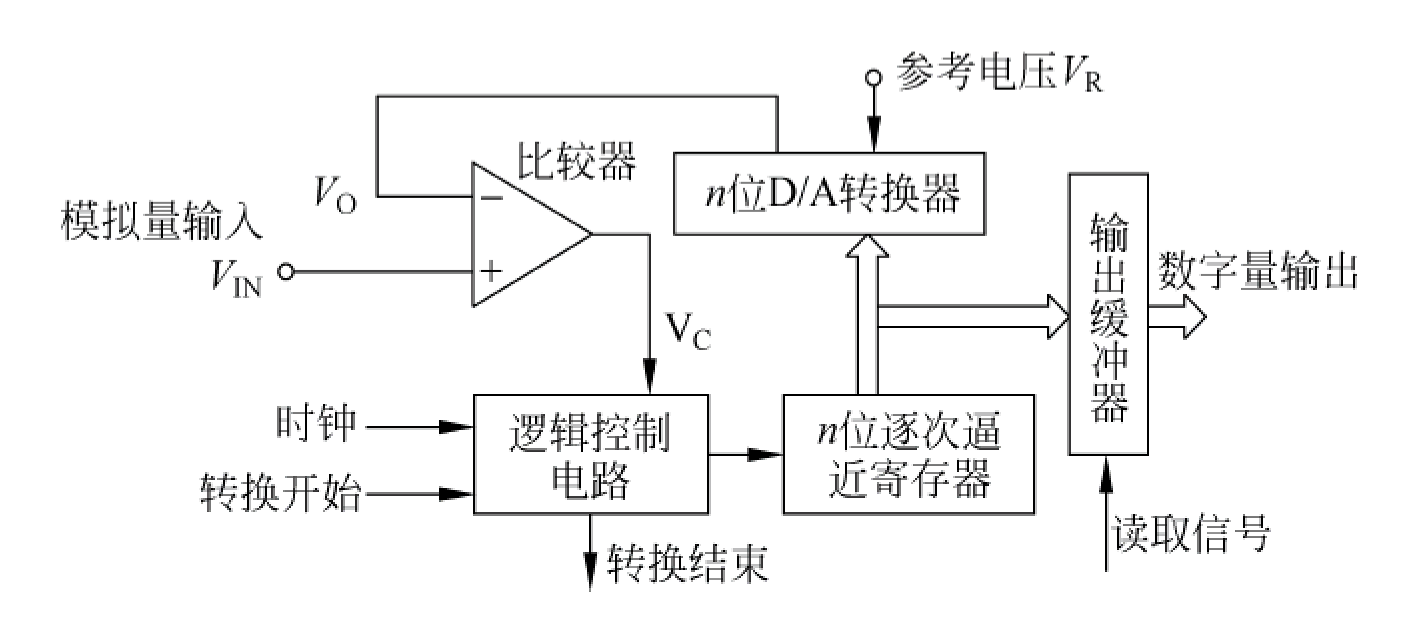


图 3.52　逐次比较型 A/D 转换原理图

一个 n 位 A/D 转换器的模数转换表达式是：

$$B=\frac{V_{IN}-V_{R-}}{V_{R+}-V_{R-}}\times 2^n \tag{3.5}$$

其中 $q=\dfrac{V_{R+}-V_{R-}}{2^n}$被称为 A/D 转换的量化单位；$n$：$n$ 位 A/D 转换器；V_{R+}、V_{R-}：基准电压源的正、负输入；V_{IN}：要转换的输入模拟量；B：转换后的输出数字量。

即当基准电压源确定之后，n 位 A/D 转换器的输出数字量 B 与要转换的输入模拟量 V_{IN}呈正比关系。例如，一个 8 位 A/D 转换器，设 $V_{R+}=5.02$V，$V_{R-}=0$V，当 V_{IN}为 2.5V 时所对应的转换数字量为 80H。

逐次比较型 A/D 转换器精度、速度较理想，适合一般模拟量输入使用，16 位以下的 A/D 普遍采用这种转换方式。常用器件有：普通型 8 位单路 ADC0801～ADC0805、8 位 8 路 ADC0808/0809、8 位 16 路 ADC0816/0817 等；混合集成高速型 12 位单路 AD574A、ADC803 等。其优点是速度较快、功耗低，在低精度(<12 位)时价格便宜，但高精度(≥12 位)时价格很高。

(4) 并行/串并行比较型

并行比较型 A/D 采用多个比较器，仅作一次比较而实行转换，又称 Flash(快速)型。由于转换速率极快，n 位的转换需要 $2n-1$ 个比较器，因此电路规模也极大，价格也高，只适用于视频 A/D 转换器等速度特别高的领域。

串并行比较型 A/D 结构上介于并行比较型和逐次比较型之间，最典型的是由两个 $n/2$ 位的并行型 AD 转换器配合 D/A 转换器组成，用两次比较实现转换，所以称为 Half Flash (半快速)型。还有分成三步或多步实现 A/D 转换的叫做分级型 A/D，而从转换时序角度

又可称为流水线型 A/D；现代的分级型 A/D 中还加入了对多次转换结果作数字运算而修正特性等功能。这类 A/D 速度比逐次比较型快，电路规模比并行型小，如 TLC5510。

(5) Σ-Δ 调制型

Σ-Δ 调制型 A/D 由积分器、比较器、1 位 D/A 转换器和数字滤波器等组成。原理上近似于积分型，将输入电压转换成时间(脉冲宽度)信号，用数字滤波器处理后得到数字值。电路的数字部分容易单片化，因此容易做到高分辨率，主要用于音频和测量领域，如 AD7705。

(6) 电容阵列逐次比较型

电容阵列逐次比较型 A/D 在内置 D/A 转换器中采用电容矩阵方式，也可称为电荷再分配型。一般的电阻阵列 D/A 转换器中多数电阻的值必须一致，在单芯片上生成高精度的电阻并不容易。如果用电容阵列取代电阻阵列，可用低成本制成高精度单片 A/D 转换器。最近的逐次比较型 A/D 转换器大多为电容阵列式的。

(7) 压频变换型

压频变换型(Voltage-Frequency Converter)是通过间接转换方式实现模数转换的。其原理是首先将输入的模拟信号转换成频率信号，然后用计数器将频率信号转换成数字量。从理论上讲这种 A/D 的分辨率几乎可以无限增加，只要采样的时间能够满足输出频率分辨率要求的累积脉冲个数的宽度。其优点是分辨率高、功耗低、价格低，但是需要外部计数电路共同完成 A/D 转换，如 AD650。

2) 典型 A/D 转换器及基本应用

在选用 A/D 转换器之前，首先需对其性能指标做一了解。A/D 转换器的性能指标主要包括以下几个方面。

(1) 分辨率

分辨率是指 A/D 转换器对微小输入信号变化的敏感程度。分辨率越高，转换时对输入量微小变化的反应越灵敏。通常用数字量的位数来表示，如 8 位、10 位、12 位等。分辨率为 n，表示它可以对满刻度的(1/2)n 的变化量做出反应。

(2) 转换速率

转换速率是指完成一次从模拟量转换到数字量的 A/D 转换所需的时间的倒数。积分型 A/D 的转换时间是毫秒级，属低速 A/D；逐次比较型 A/D 是微秒级，属中速 A/D；并行/串并行型 A/D 可达到纳秒级。而采样时间则是另外一个概念，是指两次转换的间隔。为了保证转换的正确完成，采样速率(Sample Rate)必须小于或等于转换速率。因此有人习惯上将转换速率在数值上等同于采样速率，这也是可以接受的。

(3) 转换精度

A/D 转换器的转换精度可以用绝对误差和相对误差来表示。所谓绝对误差，其误差的大小由实际模拟量输入值和理论值之差来度量；绝对误差包括增益误差、零点误差和非线性误差等。相对误差是指绝对误差与满刻度值之比，一般用百分数来表示，对 A/D 转换器常用最低有效值的位数(Least Significant Bit，LSB)来表示，$1\text{LSB}=1/2^n$。例如，对于一个 8 位 0～5V 的 A/D 转换器，如果其相对误差为 ±1LSB，则其绝对误差为 $\pm(5/2^8)\text{mV}=\pm19.5\text{mV}$，相对百分误差为 0.39%，一般来说，位数 n 越大，其相对误差(或绝对误差)越小。

下面以 ADC0808/0809 为例介绍一下 A/D 转换器的基本应用。

ADC0808/0809 为 8 位逐次逼近型 A/D 转换器，内部有 8 通道模拟开关及通道选择逻辑、8 位 A/D 转换器、三态输出锁存缓冲器等，其内部结构框图及引脚如图 3.53 所示。

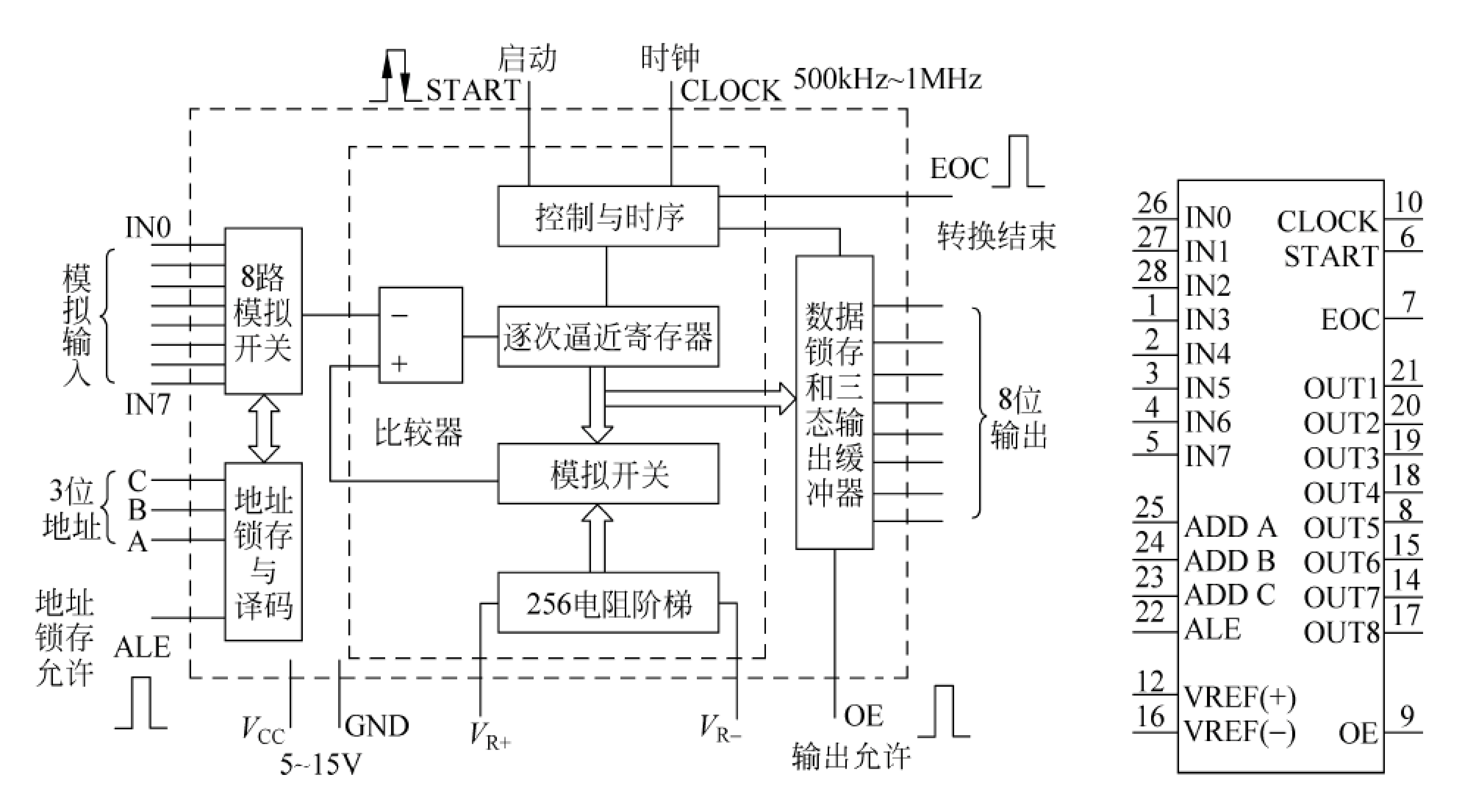


图 3.53 ADC0808/0809 内部结构框图及引脚图

ADC0808/0809 主要性能包括以下几项。

- 分辨率为 $1/2^8$≈0.39%。
- 模拟电压转换范围是 0～+5V。
- 标准转换时间为 100μs。
- 主要采用 28 脚双列直插式封装。
- 其工作时钟 CLK 为 0～640kHz，一般取 500kHz。
- 功耗为 15mW。

ADC0808/0809 各引脚功能如下。

IN0～IN7：8 路模拟量输入端，允许 8 路模拟量分时输入，共用一个 A/D 转换器。

ALE：地址锁存允许信号输入端，高电平有效，上升沿时锁存 3 位通道选择信号。

A、B、C：3 位模拟量通道选择信号线（ALE 为高电平时），与所选通道的对应关系见表 3.5。

START：启动 A/D 转换信号，输入，高电平有效。上升沿时将转换器内部清零，下降沿时启动 A/D 转换。

EOC：转换结束信号，输出，高电平有效。

OE：输出允许信号，输入，高电平有效。该信号用来打开三态输出缓冲器，将 A/D 转换得到的 8 位数字量送到数据总线上。

D0～D7：8 位数字量输出，由于有三态输出锁存，可与主机数据总线直接相连。

CLOCK：外部时钟脉冲输入端，当脉冲频率为 640kHz 时，A/D 转换时间为 100μs。

V_{R+}、V_{R-}：基准电压源正、负端。取决于被转换的模拟电压范围，通常 V_{R+} = +5V DC，V_{R-} = 0V DC。

Vcc：工作电源，+5V DC。

GND：电源地。

表 3.5　模拟量通道选择逻辑

C	B	A	选择通道
0	0	0	IN0
0	0	1	IN1
0	1	0	IN2
0	1	1	IN3
1	0	0	IN4
1	0	1	IN5
1	1	0	IN6
1	1	1	IN7

A/D转换的基本步骤为：首先ALE的上升沿将地址代码锁存，译码后选通模拟开关中的某一路，使该路模拟量进入到A/D转换器中；然后，START的上升沿将转换器内部清零，下降沿起动A/D转换，即在时钟的作用下，逐位逼近过程开始，转换结束信号EOC即变为低电平；当转换结束后，EOC恢复高电平；此时，如果对输出允许OE输入一高电平命令，则可读出数据。ADC0808/0809基本功能可用Proteus软件进行仿真，仿真电路如图3.54所示。

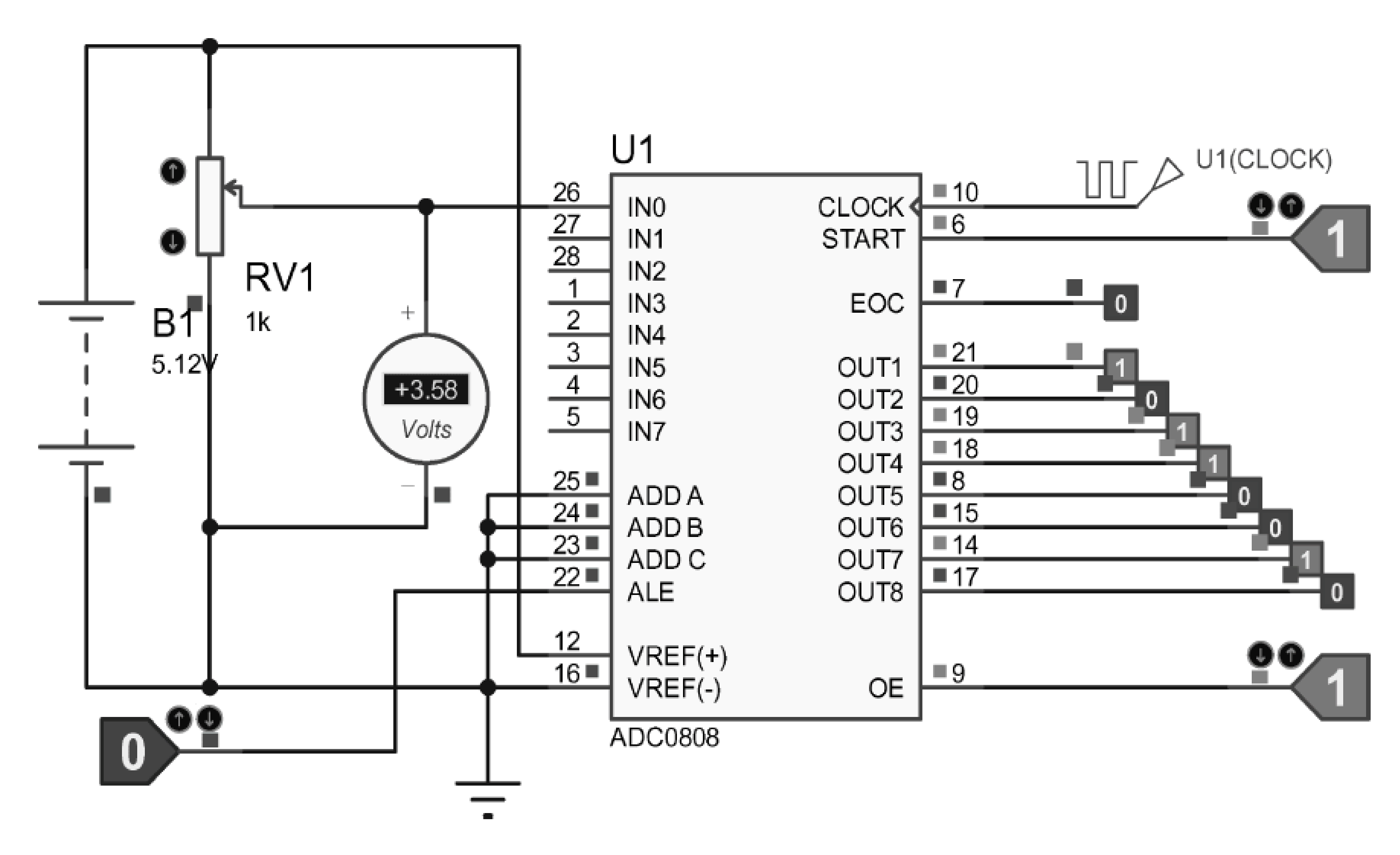


图 3.54　ADC0808 仿真电路

A/D转换器的接口电路主要是解决主机如何分时采集多路模拟量输入信号的，即主机如何启动A/D转换，如何判断A/D完成一次模数转换，如何读入并存放转换结果的。典型A/D转换器接口电路根据查询转换结果的方式不同，主要有查询方式、定时方式和中断方式三种。基于ADC0809的查询方式基本应用电路见图3.55。

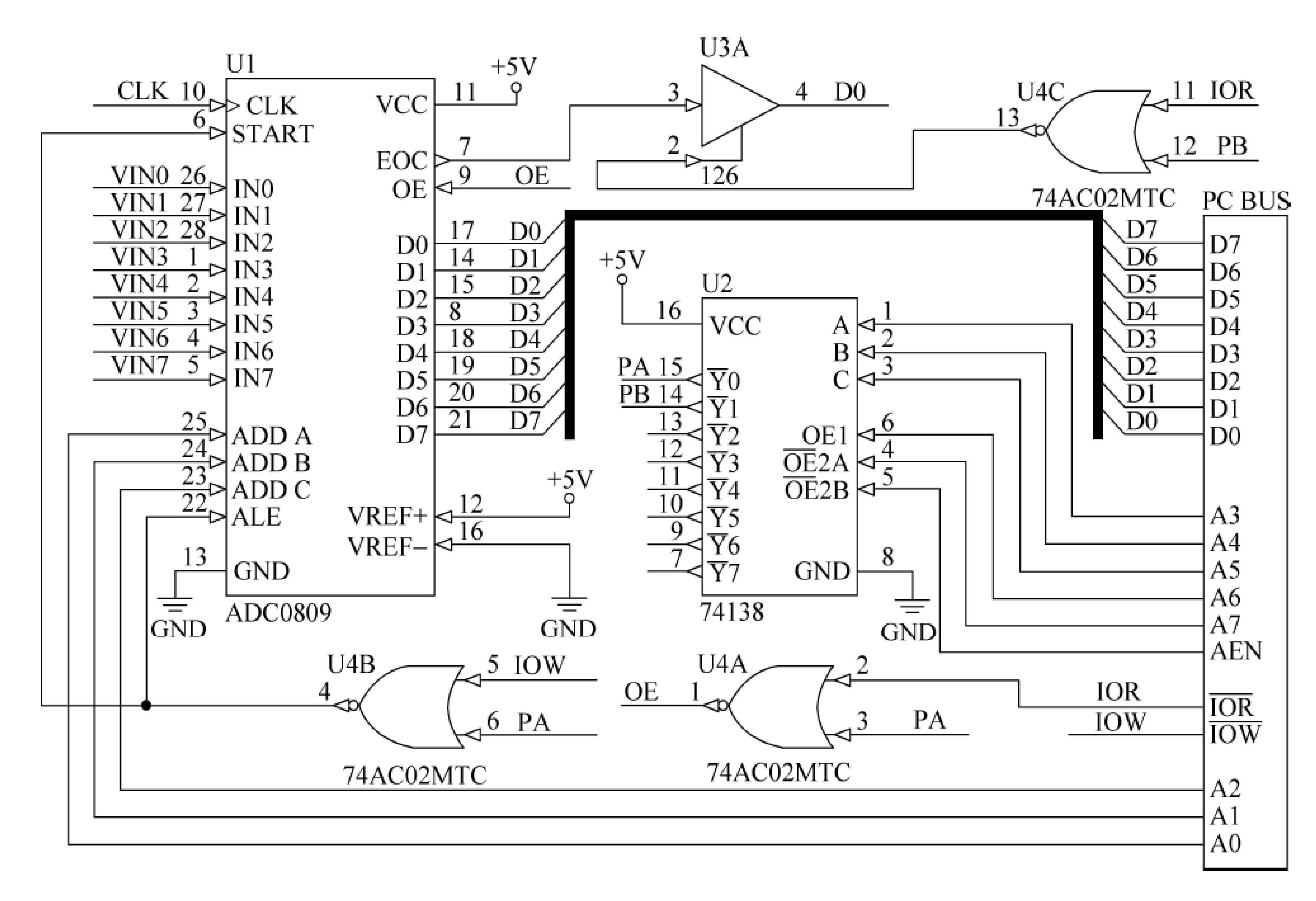


图 3.55　基于 ADC0809 的查询方式基本应用电路

接口程序如下：

```
        MOV   BX,BUFF   ; 置采样数据区首址
        MOV   CX,08H    ; 8 路输入
START:  OUT   PA,AL     ; 启动 A/D 转换
REOC:   IN    AL,PB     ; 读 EOC
        RCR   AL,01     ; 判断 EOC
        JNC   REOC      ; 若 EOC = 0,继续查询
        IN    AL,PA     ; 若 EOC = 1,读 A/D 转换数
        MOV   [BX],AL   ; 存 A/D 转换数
        INC   BX        ; 存 A/D 转换数地址加 1
        INC   PA        ; 接口地址加 1
        LOOP  START     ; 循环
```

下面简要说明 A/D 转换过程。

（1）首先主机执行一条启动转换 VIN0 的输出指令，即把 AL 中的数据送到地址为 PA 的接口电路中，此时 AL 中的内容无关紧要，地址 PA＝40H 使 38 译码器输出一个低电平，连同 OUT 指令造成的低电平，使或非门 U4B 产生脉冲信号到引脚 ALE 和 START，ALE 的上升沿将通道地址代码 000 锁存并进行译码，选通模拟开关中的 VIN0，使该路模拟量进入 A/D 转换器；同时 START 的上升沿将 ADC0809 中的逐次逼近寄存器 SAR 清零，下降沿启动 A/D 转换，即在时钟的作用下，逐次逼近的模数转换过程开始。

（2）主机查询转换结束信号 EOC 的状态，通过执行输入指令，即把地址为 PB 的接口电路的数据读入 AL 中，此时地址 PB＝48H，使 38 译码器输出一个低电平，连同 IN 指令造成的低电平，使或非门 U4C 产生脉冲信号并选通 126 三态缓冲器，使 EOC 出现在数据线 D0

上。然后将读入的8位数据进行带进位循环右移，以判断EOC的电平状态。如果EOC为“0”，表示A/D转换正在进行，程序再跳回REOC，反复查询；当EOC为“1”，表示A/D转换结束。

(3) 然后，主机便执行一条输入指令，把接口地址为PA的转换数据读入AL中，即输出一个低电平，连同IN指令造成的低电平，使或非门U4A产生脉冲信号，即产生输出允许信号到OE，使ADC0809内部的三态输出锁存器释放转换数据到数据线上，并被读入到AL中。

(4) 接下来，把A/D转换数据存入寄存器BX所指的数据区首地址0000H中，数据区地址加1，为第2路A/D转换数据的存放作准备；接口地址PA加1，准备接通第2路模拟量信号；计数器减1，不为0则返回到START，继续进行下一路的A/D转换。如此循环，直至完成8路A/D转换。

3) 其他A/D转换器

对于其他类型的A/D转换器，读者可根据以上基本理论，通过网上查询器件的数据手册自行了解使用。例如高性能、12位逐次逼近型A/D转换器AD574A，其内部结构大体与ADC0809类似，由12位A/D转换器、控制逻辑、三态输出锁存缓冲器与10V基准电压源构成，可以直接与主机数据总线连接，但只能输入一路模拟量，采用28脚双立直插式封装。采用AD574A的模拟量输入接口电路如图3.56所示：8通道数字控制模拟电子开关CD4051作为多路选择器，选中模拟量V_0～V_7中一路输入，经采样保持器LF398送AD574A进行A/D转换，其中8255A是可编程并行I/O接口芯片，A/D转换结果通过8255送PC总线，从而送CPU进行数据处理。

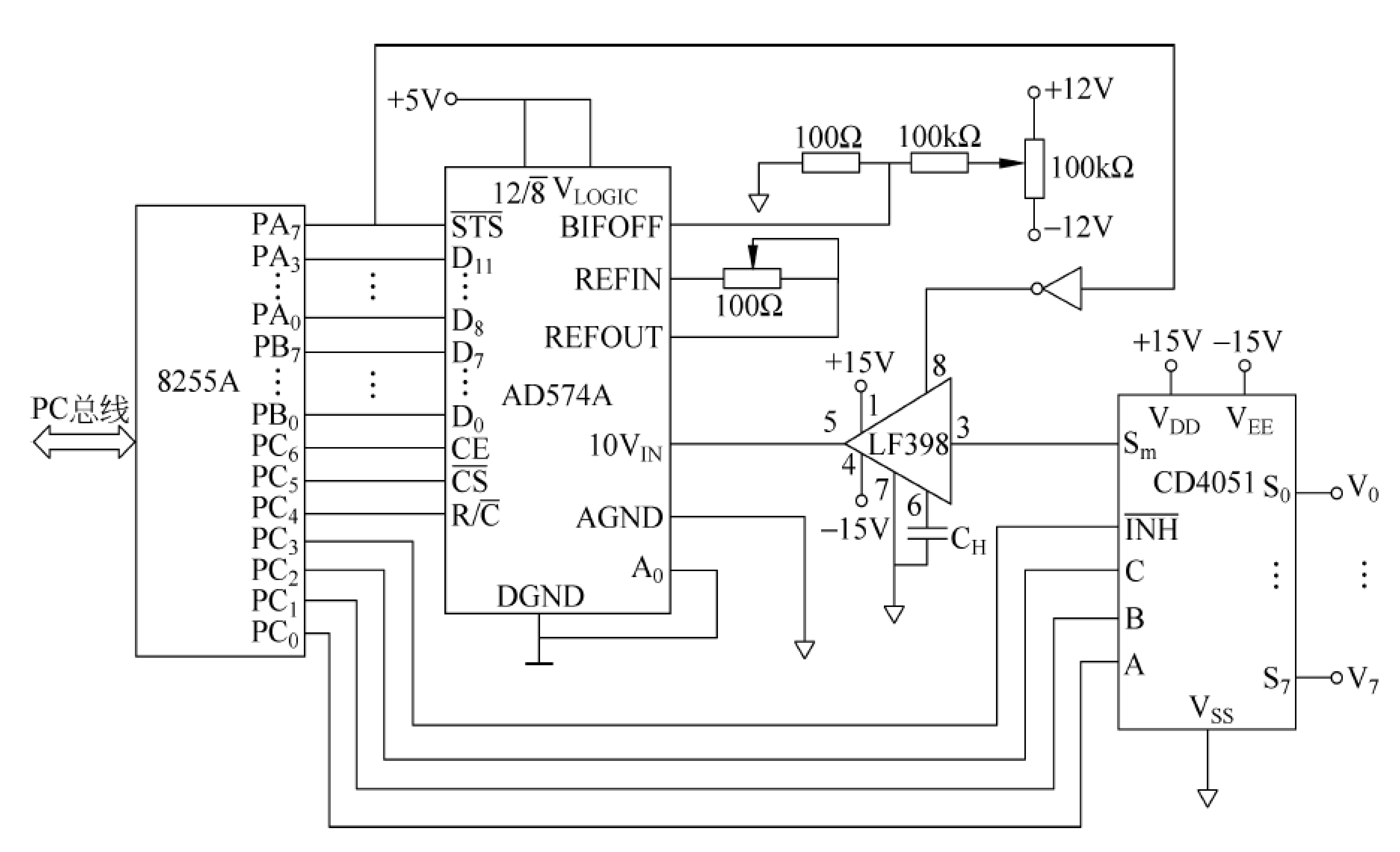


图3.56　采用AD574A的模拟量输入接口电路

3.4.2　模拟量输出通道

模拟量输出通道是计算机控制系统实现对生产过程控制的关键，它主要将计算机处理后输出的数字信号转换成模拟信号去驱动相应的模拟量执行机构，以达到一定的控制任务，

例如对电动调节阀的控制等。

模拟量输出通道又称 D/A 通道或 AO 通道，一般由输出接口、数/模转换器(D/A 或 DAC)、电压/电流转换(V/I)电路、驱动电路、执行机构等 5 部分组成。另外，在输出接口与 D/A 转换之间，可进行信号的光电隔离。D/A 转换器是 AO 通道的核心器件，结构图如图 3.40 所示。AO 通道结构形式可根据 D/A 转换器的选用数量分为两类：一类是每一个通道设置一个 D/A 转换器，D/A 转换器芯片内部一般都带有数据锁存器，采用数字量保持方案，即 D/A 转换器兼顾数字信号转换成模拟信号及信号保持的作用，其结构简单，转换速度快，工作可靠，精度较高、通道独立，缺点是所需 D/A 转换器芯片较多，如图 3.57(a)所示；另一类是多个通道共用一个 D/A 转换器，但每一通道都配有一个采样保持放大器，D/A 转换器只起数字到模拟信号的转换作用，采样保持器用来实现模拟信号保持功能，即采用模拟量保持方案，节省 D/A 转换器，但电路复杂，精度差，可靠低、占用主机时间，常用于通道数量多，速度要求不高的场合，如图 3.57(b)所示。

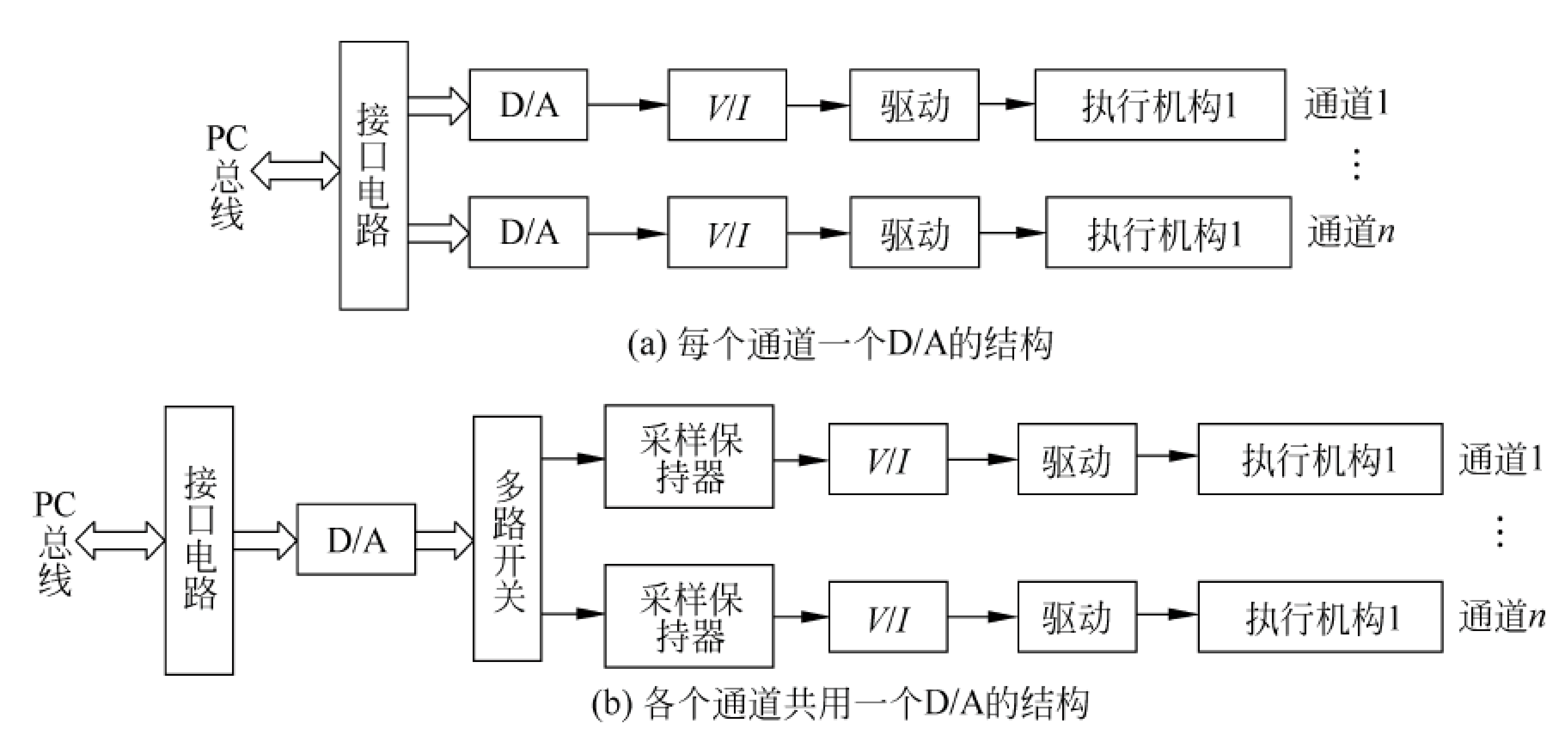


图 3.57 模拟量输出通道的基本结构

下面主要针对 D/A 转换和 V/I 转换电路进行介绍。

1. D/A 转换

D/A 转换指将数字信号转变为模拟信号，D/A 转换器(DAC)是将输入数字信号转换成模拟信号的装置。

1) D/A 转换的基本原理

D/A 转换的基本原理是：将输入的每一位二进制代码按其权的大小转换成相应的模拟量，然后将代表各位的模拟量相加，所得的总模拟量与数字量成正比，这样便实现了从数字量到模拟量的转换。常见的 D/A 转换方式有：权电阻式 DAC、T 型电阻网络 DAC、权电流型 DAC、权电容型 DAC、开关树型 DAC 等。下面以 4 位 D/A 转换器为例说明其工作原理。如图 3.58 所示为权电阻式 D/A 转换电路，图中 d_3, d_2, d_1, d_0 表示 4 位二进制数的各个位，以 d_0 为例，d_0 为“1”时，表示开关 S0 接通右侧，为“0”时，接通左侧，根据电路原理，可得：

$$\begin{aligned} I &= I_0 d_0 + I_1 d_1 + I_2 d_2 + I_3 d_3 \\ &= \frac{V_{\mathrm{REF}}}{8R} d_0 + \frac{V_{\mathrm{REF}}}{4R} d_1 + \frac{V_{\mathrm{REF}}}{2R} d_2 + \frac{V_{\mathrm{REF}}}{R} d_3 = \frac{V_{\mathrm{REF}}}{2^3 R}(d_3 \cdot 2^3 + d_2 \cdot 2^2 + d_1 \cdot 2^1 + d_0 \cdot 2^0) \end{aligned}$$

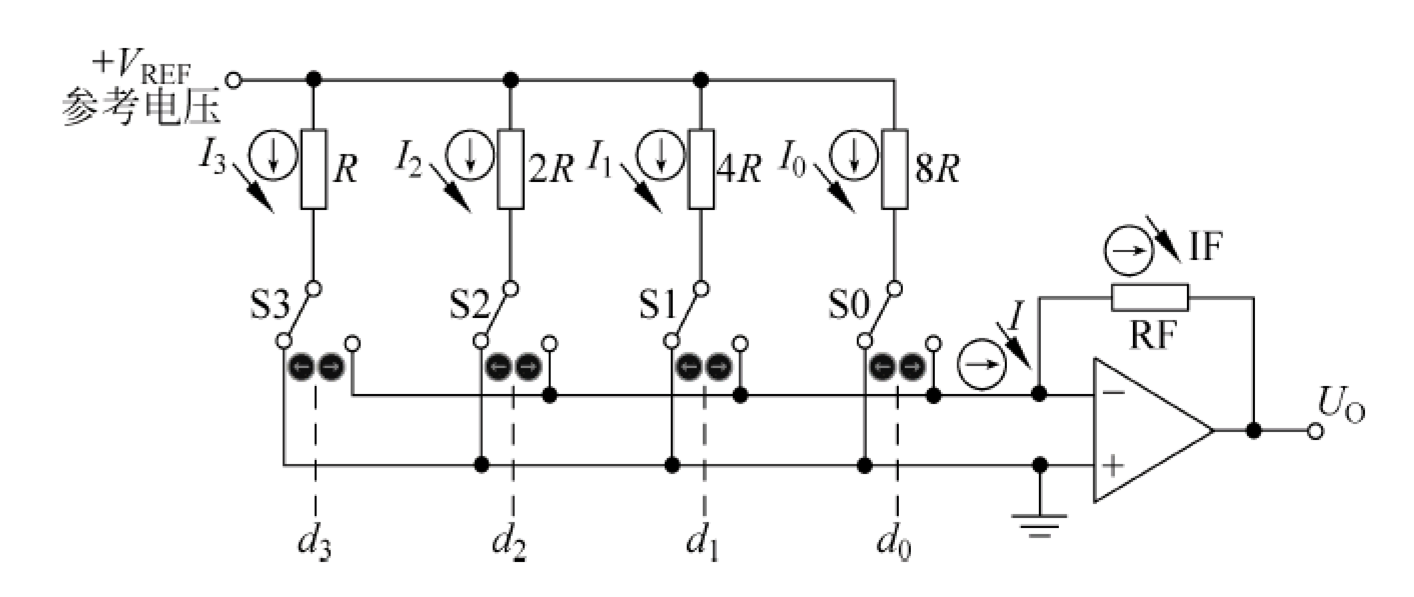


图 3.58 权电阻式 D/A 转换电路

因此：

$$U_O = K_u(d_{n-1} \cdot 2^{n-1} + d_{n-2} \cdot 2^{n-2} + \cdots + d_1 \cdot 2^1 + d_0 \cdot 2^0)$$

即将输入的数字信号 d_3, d_2, d_1, d_0 转变成了模拟信号 U_O。由上述推导可见，输出电压除了与输入的二进制数有关，还与运算放大器的反馈电阻 RF 以及基准电压 V_{REF} 有关。

2）典型 D/A 转换器及基本应用

首先了解一下 D/A 转换器的性能指标，这是衡量芯片质量的重要参数，也是选用 D/A 芯片型号的依据。其主要性能指标如下。

（1）分辨率

分辨率是指 D/A 转换器能分辨的最小输出模拟增量，即当输入数字量发生单位数码变化时所对应输出模拟量的变化量，它取决于能转换的数字量二进制位数，位数越多，分辨率也就越高。其分辨率与二进制位数 n 呈下列关系：

$$\text{分辨率} = \text{满刻度值}/2^n = V_{REF}/2^n$$

（2）转换精度

转换精度分绝对精度和相对精度，是由于非线性、零点刻度、满量程刻度及温漂等因素引起的误差。精度表示 D/A 转换器实际输出与其理论值的误差。对于分辨率很高的 D/A 转换器并不一定具有很高的精度。

（3）偏移量误差

偏移量误差是指输入数字量时，输出模拟量对于零的偏移值。此误差可通过 D/A 转换器的外接 V_{REF} 和电位器加以调整。

（4）稳定时间

稳定时间是用来表示 D/A 转换速度的一个参数，指从输入数字量变换到输出模拟量达到终值误差 1/2LSB 时所需的时间。显然，稳定时间越大，转换速度越低。对于输出是电流的 D/A 转换器来说，稳定时间是很快的，约几微秒，而输出是电压的 D/A 转换器，其稳定时间主要取决于运算放大器的响应时间。

（5）线性度

线性度是指 DAC 的实际转换特性曲线和理想直线之间的最大偏差。通常，线性度不应超过±1/2LSB。

（6）输出极性与范围

输出极性有单极性与双极性两种。其输出范围与参考电压有关，对于电流输出的 D/A 转换器，要用运算放大器将其转为电压，因而输出范围与转换电路也有关。

根据 D/A 转换器的分辨率，D/A 转换器有 8 位、10 位、12 位、14 位之分；另外有 TTL 工艺的和 CMOS 工艺的；有并行 D/A 转换器（传统芯片，如 AD1408、DAC100、AD7532、AD7541、DAC0808、DAC0832、5G7520 等），也有串行 D/A 转换器（如 MAX518 等）。下面以 DAC0832 为例介绍一下 D/A 转换器的基本应用。

DAC0832 是一个 8 位的，含有双输入数据锁存器的数/模转换器，内部结构原理框图和引脚如图 3.59 所示，主要特点有以下几个方面。

- 8 位 D/A 转换器。
- CMOS 工艺。
- 倒 T 型电阻网络。
- 内部有两个数据寄存器。
- 有直通、单缓冲、双缓冲三种工作方式。

如图 3.59 中，DAC0832 主要由 8 位输入寄存器、8 位 DAC 寄存器、8 位 D/A 转换器以及输入控制电路四部分组成。8 位输入寄存器用于存放主机送来的数字量，使输入数字量得到缓冲和锁存，并加以控制；8 位 DAC 寄存器用于存放待转换的数字量，并加以控制；8 位 D/A 转换器输出与数字量成正比的模拟电流；由与门、非与门组成的输入控制电路用来控制两个寄存器的选通或锁存状态。$\overline{LE1}=1$，数据直通，寄存器输出随输入变化；$\overline{LE1}=0$，数据锁存，寄存器输出不随输入变化；$\overline{LE2}$同样。例如，当$\overline{WR2}$和$\overline{XFER}$同时有效时，$\overline{LE2}$为高电平“1”，此时 DAC 寄存器的输出端 Q 跟随输入端 D 也就是输入寄存器 Q 端的电平变化；反之，当$\overline{LE2}$为低电平“0”时，第一级 8 位输入寄存器 Q 端的状态则锁存到第二级 8 位 DAC 寄存器中，以便第三级 8 位 DAC 转换器进行 D/A 转换。

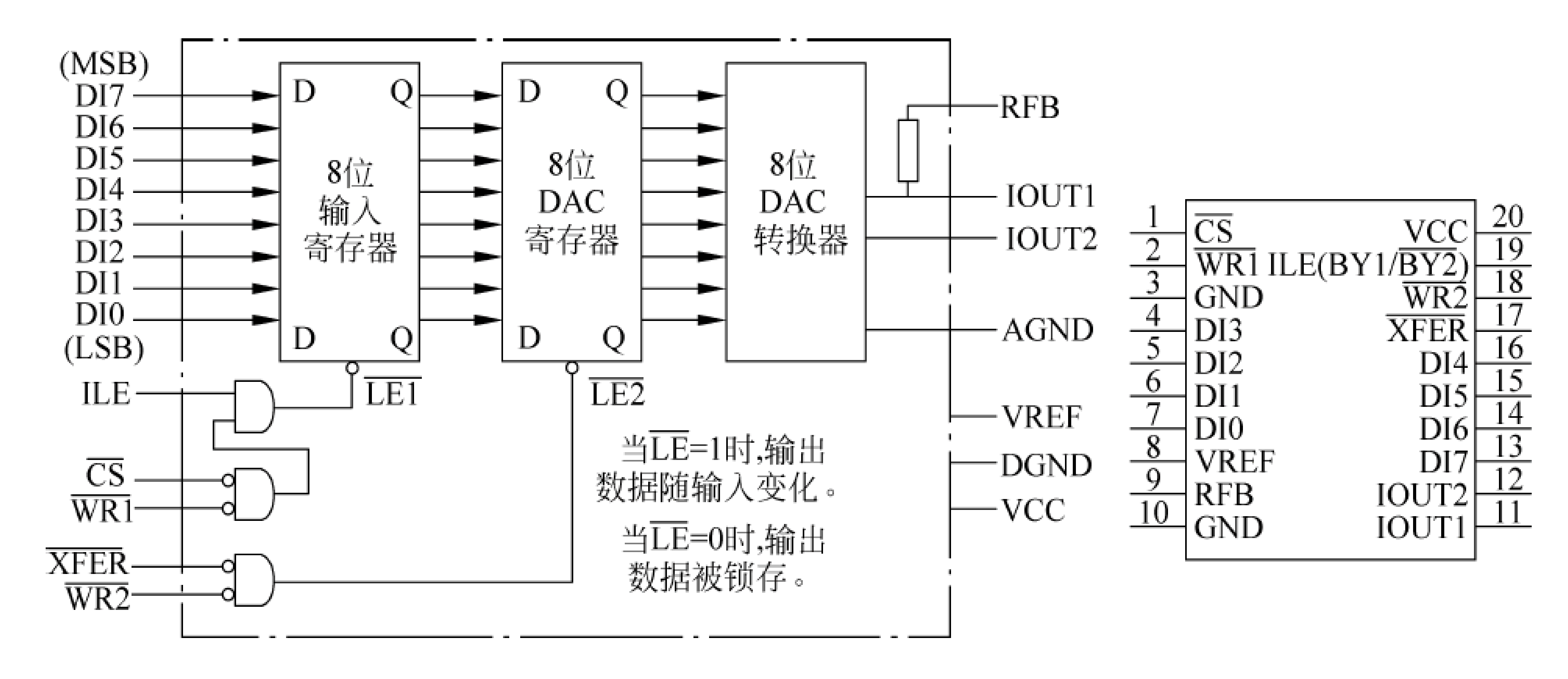


图 3.59　DAC0832 内部结构框图

DAC0832 各引脚功能如下。

DI0～DI7：输入 8 位数字量，DI7 最高位（MSB），DI0 最低位（LSB）。

VREF：基准电压，VREF＝－10～＋10V。

VCC：电源电压，＋5～＋15V。

IOUT1、IOUT2：模拟电流输出端，IOUT1＋IOUT2＝常数＝VREF/RFB，输入全 1 时，IOUT1 最大，IOUT2 最小；反之则反。

RFB：运放用反馈电阻引出端。

AGND：第 3 引脚，模拟信号接地端。

DGND：第 10 引脚，数字信号接地端。

$\overline{\text{CS}}$：片选信号，低电平有效。

$\overline{\text{WR1}}$：写信号 1，低电平有效。

$\overline{\text{WR2}}$：写信号 2，低电平有效。

ILE：输入锁存使能，高电平有效。

$\overline{\text{XFER}}$：DAC 转换控制端，低电平有效。

由于 DAC0832 的输出是电流形式，所以在其电流输出端需接一个运算放大器，将输出变为电压信号，RFB 端接运放输出端。VREF 接－5V 时，电压输出为 0～＋5V；VREF 接－10V，电压输出为 0～＋10V；VREF 接＋5V，电压输出为 0～－5V；VREF 接＋10V，电压输出为 0～－10V。

下面针对 DAC0832 的直通、单缓冲、双缓冲三种工作方式，以及其他需要注意的问题作进一步的了解。

(1) 直通工作方式

所谓直通工作方式就是数据不作任何锁存，使 ILE、$\overline{\text{CS}}$、$\overline{\text{WR1}}$、$\overline{\text{WR2}}$、$\overline{\text{XFER}}$始终处于有效状态，使得$\overline{\text{LE1}}$、$\overline{\text{LE2}}$始终处于高电平“1”，两级缓冲器都处于选通状态，如图 3.60 为使用 Proteus 软件仿真情况。图中输入数字量为 FFH，在未做任何校正的情况下，输出为 4.94V。

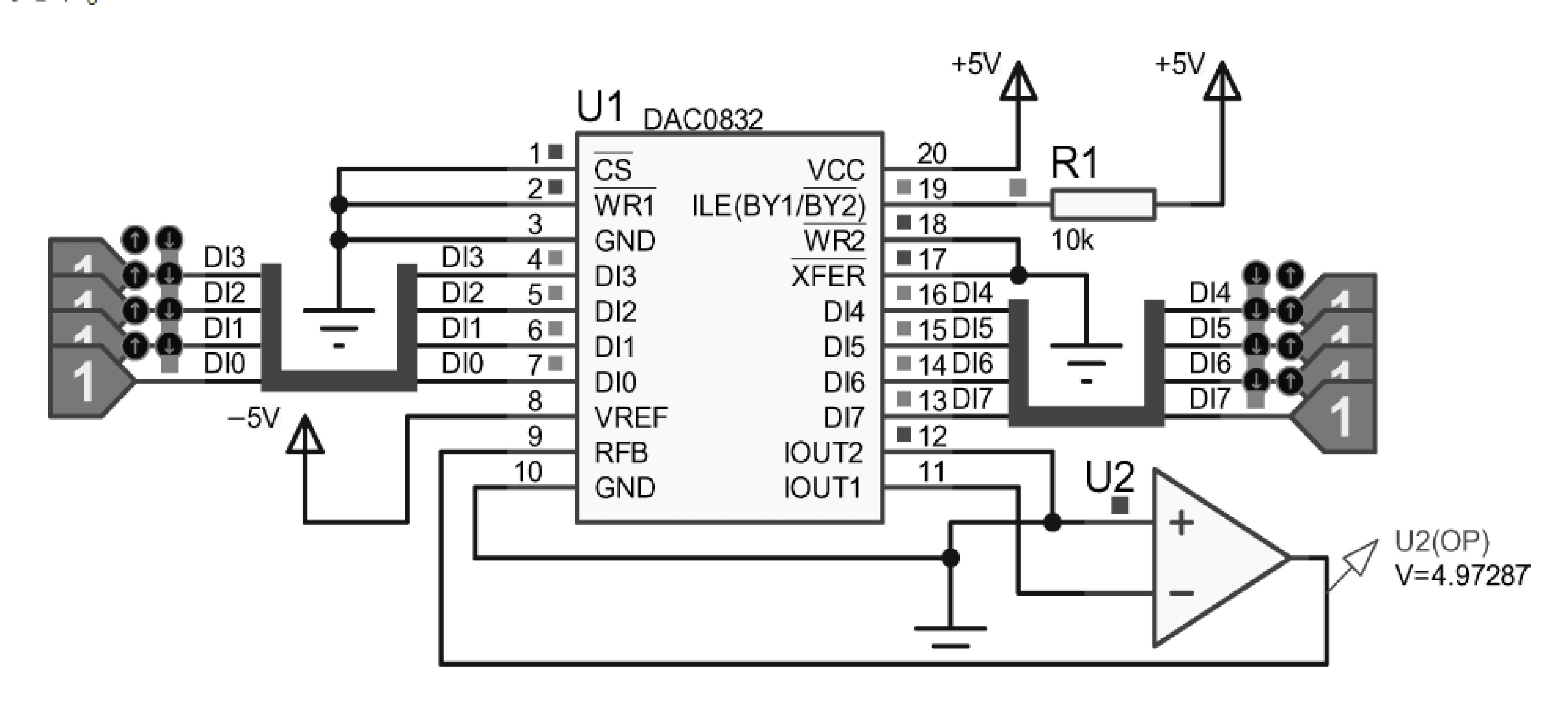


图 3.60　DAC0832 的直通工作方式

(2) 单缓冲工作方式

所谓单缓冲方式就是数据被一个寄存器锁存，即输入寄存器工作于受控状态，DAC 寄存器工作于直通状态，如图 3.61 所示。

(3) 双缓冲工作方式

双缓冲方式就是两个寄存器都对数据进行锁存的工作方式，也就是两个寄存器都工作于受控状态。可以在图 3.61 的基础上做一些修改即可得到，如图 3.62 所示。

(4) 调零和调满度

当 DAC0832 的转换精度不满足要求，存在较大误差时，可对其进行调零和调满度，如图 3.63，具体方法如下。

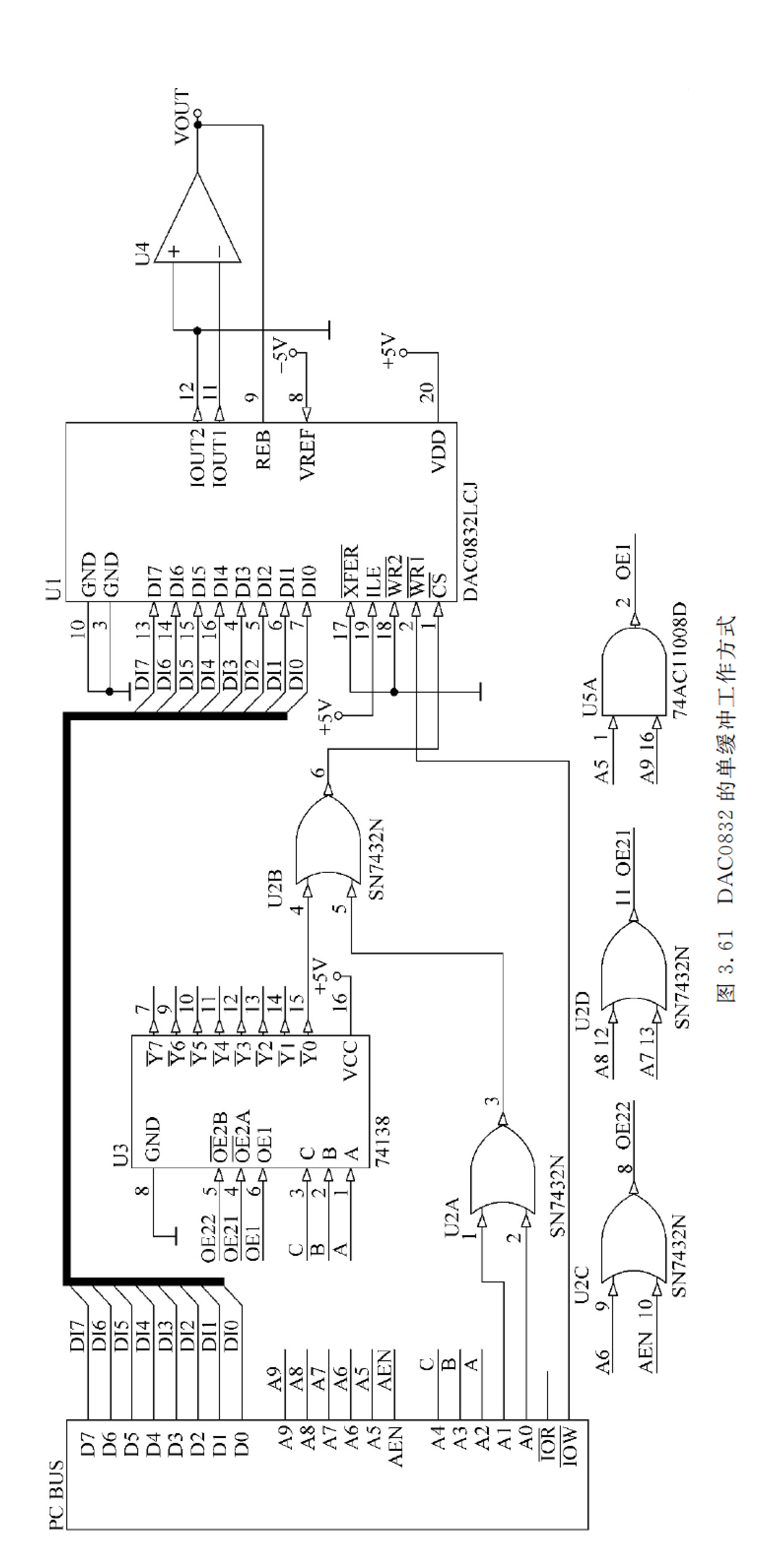


图 3.61 DAC0832 的单缓冲工作方式

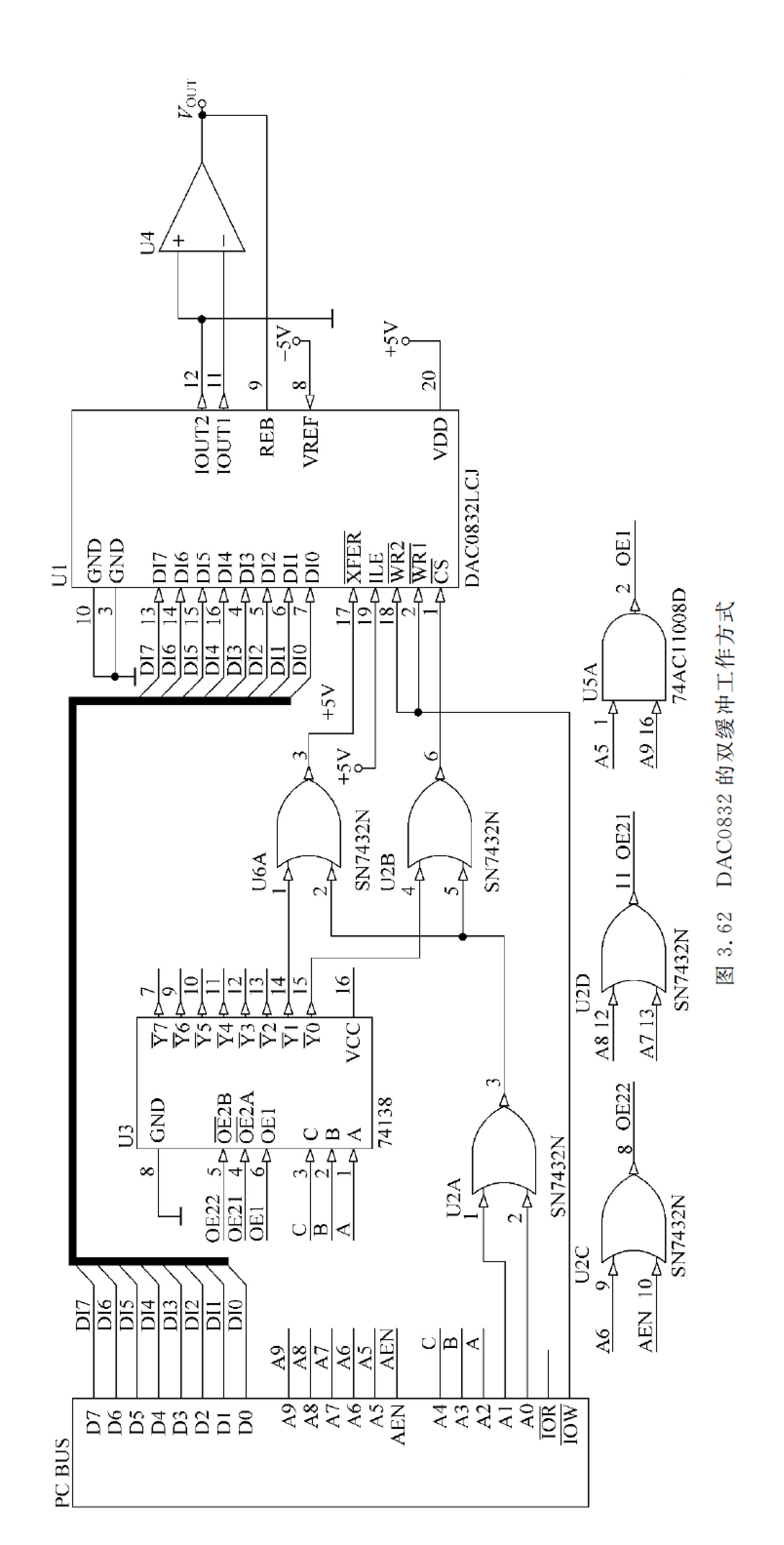


图 3.62 DAC0832 的双缓冲工作方式

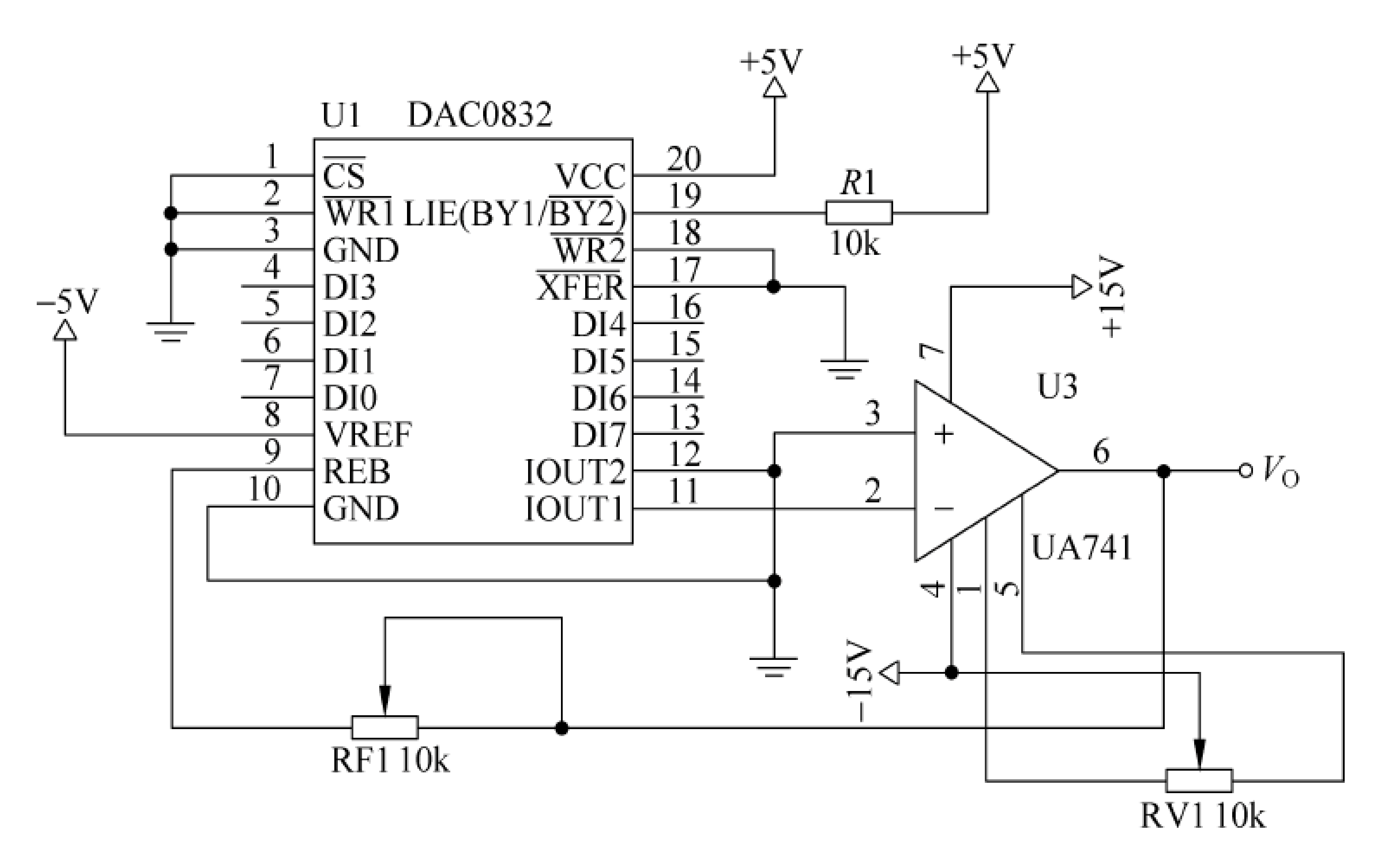


图 3.63　DAC0832 的调零和调满度电路

① 调零：令输入 DI0～DI7 为全零，调节 RF1 使 $V_O=0$。

② 调满度：输入 DI0～DI7 为全 1，调 RV1 使 V_O＝最大。

③ 重复①、②步骤，直到都满足为止。

3）单极性与双极性电压输出电路

如上所述，通常采用 D/A 转换器外加运算放大器的方法将 DAC 的电流输出变为电压输出，输出电路包括单极性输出和双极性输出两种情况。如图 3.64 所示，V_{OUT1} 为单极性输出，V_{OUT2} 为双极性输出。设 D 为 n 位 DAC 输入数字量，V_{REF} 为参考电压，则有：

单极性输出：$V_{OUT1}=-V_{REF}\cdot\frac{D}{2^n}, D=D_7\times2^7+D_6\times2^6+\cdots+D_1\times2^1+D_0\times2^0$

双极性输出：$V_{OUT2}=-\left(\frac{R_3}{R_1}V_{REF}+\frac{R_3}{R_2}V_{OUT1}\right)=V_{REF}\left(\frac{D}{2^{n-1}}-1\right)$

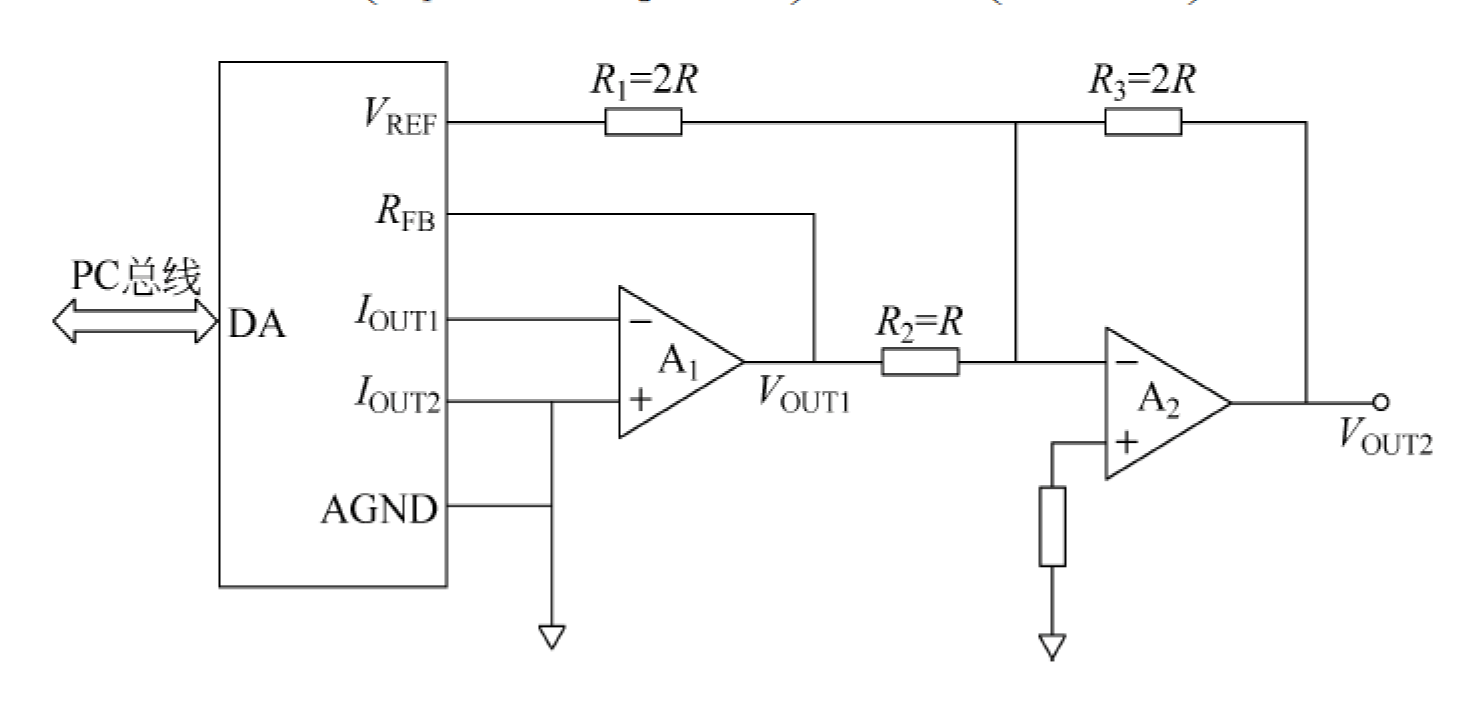


图 3.64　DAC 单极性与双极性电压输出电路

4）其他 D/A 转换器

D/A 转换器还有许多其他类型、不同公司的产品，例如通用、廉价的 DAC 有 AD1408、AD7524、AD558；高速、高精度 DAC：AD562、AD7541；高速 DAC：AD561、DAC-08；高分辨率 DAC：DAC1136、DAC1137 等。可以通过查询器件的数据手册作进一步了解。

下面对 DAC1210 做一简单介绍。DAC1210 是一个 12 位 D/A 转换器，电流输出方式，其结构原理与控制信号功能基本类似于 DAC0832。由于它比 DAC0832 多了 4 条数据输入线，故有 24 条引脚，DAC1210 内部原理框图如图 3.65 所示，内部有三个寄存器：一个 8 位输入寄存器，用于存放 12 位数字量中的高 8 位 DI_{11}～DI_4；一个 4 位输入寄存器，用于存放 12 位数字量中的低 4 位 DI_3～DI_0；一个 12 位 DAC 寄存器，存放上述两个输入寄存器送来的 12 位数字量；12 位 D/A 转换器用于完成 12 位数字量的转换。

DAC1210 进行 D/A 转换的基本工作原理为：当 $\overline{CS}$、$\overline{WR_1}$ 为低电平“0”，$BYTE_1/\overline{BYTE_2}$ 为高电平“1”时，$\overline{LE1}$、$\overline{LE2}$ 为“1”，选通 8 位和 4 位两个输入寄存器，将要转换的12 位数据全部送入寄存器；当 $BYTE_1/\overline{BYTE_2}$ 为低电平“0”时，$\overline{LE1}$ 为“0”，8 位输入寄存器锁存刚传送的 8 位数据，而 $\overline{LE2}$ 仍为“1”，4 位输入寄存器仍为选通，新的低 4 位数据将刷新刚传送的 4 位数据。因此，在与计算机接口电路中，计算机必须先送高 8 位后送低 4 位。$\overline{XFER}$（传送控制信号、低电平有效）和 $\overline{WR_2}$（写信号、低电平有效）用来控制 12 位 DAC 寄存器，当 $\overline{XFER}$ 和 $\overline{WR_2}$ 同为低电平“0”时，与门输出 $\overline{LE3}$ 为“1”，12 位数据全部送入 DAC 寄存器；当有一个为高电平“1”时，与门输出 $\overline{LE3}$ 即为“0”，则 12 位 DAC 寄存器锁存住数据使 12 位 D/A 转换器开始数模转换。

如图 3.65 为 DAC1210 基本应用电路。由 DAC1210 转换芯片、运算放大器以及地址译码电路组成。与 8 位 DAC0832 接口电路不同的是，除了数据总线 D7～D0 与 DAC1210 高 8 位 DI_{11}～DI_4 直接相连，D3～D0 还要与 DAC1210 低 4 位 DI_3～DI_0 复用，因而控制电路也略为复杂。在软件设计中，为了实现 8 位数据线 D0～D7 传送 12 位被转换数，主机须分两次传送被转换数。首先将被转换数的高 8 位传给 8 位输入寄存器 DI_{11}～DI_4，再将低 4 位传给 4 位输入寄存器 DI_3～DI_0，然后再打开 DAC 寄存器，把 12 位数据送到 12 位 D/A 转换器去转换。当输出指令执行完后，DAC 寄存器又自动处于锁存状态以保持数模转换的输出不变。设 12 位被转换数的高 8 位存放在 DATA 单元中，低 4 位存放在 DATA＋1 单元中，三个口地址：低 4 位输入寄存器为 380H，高 8 位输入寄存器为 381H，12 位 DAC 寄存器为 384H。可参考如下转换程序。

```
DAC: MOV  DX,0381H;
     MOV  AL,[DATA];
     OUT  DX,AL      ; 送高 8 位数据
     DEC  DX
     MOV  AL,[DATA + 1]
     OUT  DX,AL      ; 送低 4 位数据
     MOV  DX,0384H
     OUT  DX,AL      ; 完成 12 位数据转换
```

其同系列芯片 DAC1208、DAC1209 也可以相互代换使用。

2. *V*/*I* 转换电路

因为电流信号易于远距离传送，且不易受干扰，所以在计算机控制系统输出通道中常以电流信号来传送信息，这就需要将 D/A 转换电路得到的电压信号再转换成电流信号，这种完成电流输出方式的电路称为 *V*/*I* 变换电路。*V*/*I* 转换电路一般有两种形式：一是利用普

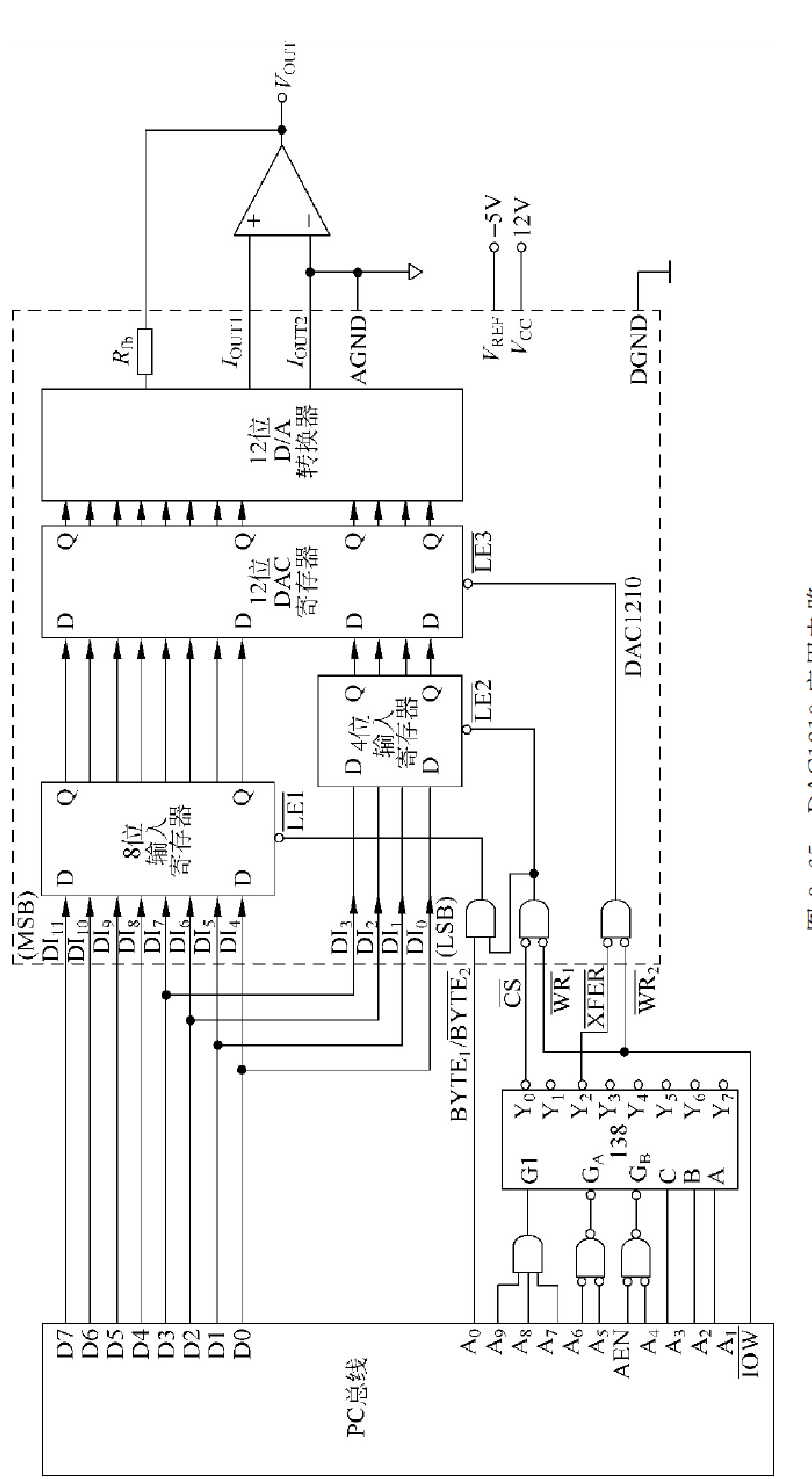


图 3.65 DAC1210 应用电路

通运放搭建功能电路；二是采用集成 V/I 转换器。下面主要以集成 V/I 转换器 ZF2B20 为例介绍一下 V/I 转换器的应用。

ZF2B20 是一个完整的电压/电流变换器，可用于子系统之间的信息传递，广泛用于工业仪表和控制系统，如发射机、显示器、控制器、记录仪、计算机、驱动器、信号控制器等一些操作控制单元之间的连接。输入电压为 0～10V；输出电流为 0～10mA 或 4～20mA；电源电压可接 10～32V。

ZF2B20 中包括一个高精度运算放大器、精密电阻和一个高稳定度的基准电压源。其内部基准电压源可输出一个 2.5V 的基准电压（第 4 引脚 REFOUT 为输出端），目的是扩大偏置和输出能力。当 REFIN 和 REFOUT 连接时，对应 0V 输入电压可以相应输出 4mA 电流。如图 3.66 是集成 V/I 转换器 ZF2B20 的引脚及基本接线图，采用单正电源供电，ZF2B20 的输入电阻为 10kΩ，动态响应时间小于 25μs，非线性小于±0.025％。

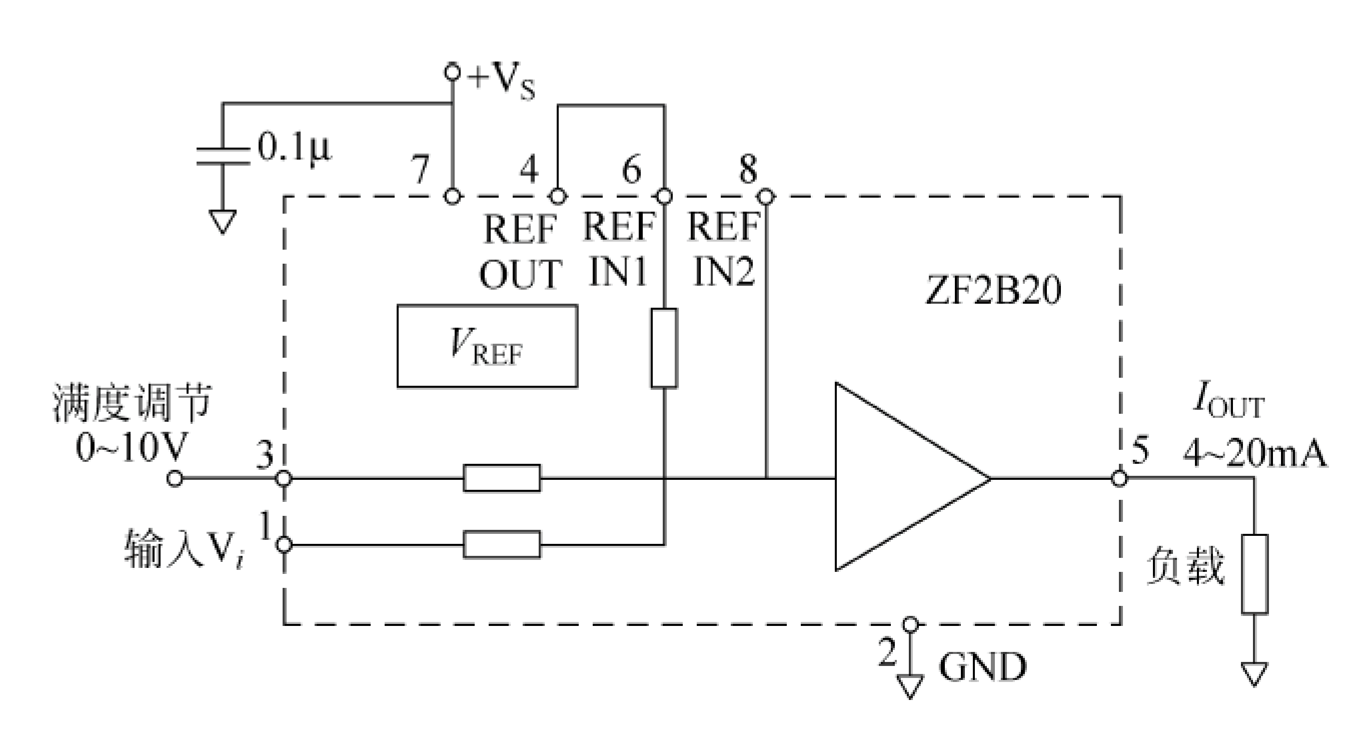


图 3.66 ZF2B20 的引脚及基本接线图

通过 ZF2B20 可以产生一个与输入电压成比例的输出电流，它的特点是低漂移，在工作温度为－25～85℃范围内，最大温漂为 0.005％/℃。利用 ZF2B20 实现 V/I 转换的电路非常简单，图 3.67(a)所示电路是一种带初值校准的 0～10V 到 4～20mA 的转换电路；图 3.67(b)则是一种带满度校准的 0～10V 到 0～10mA 的转换电路。

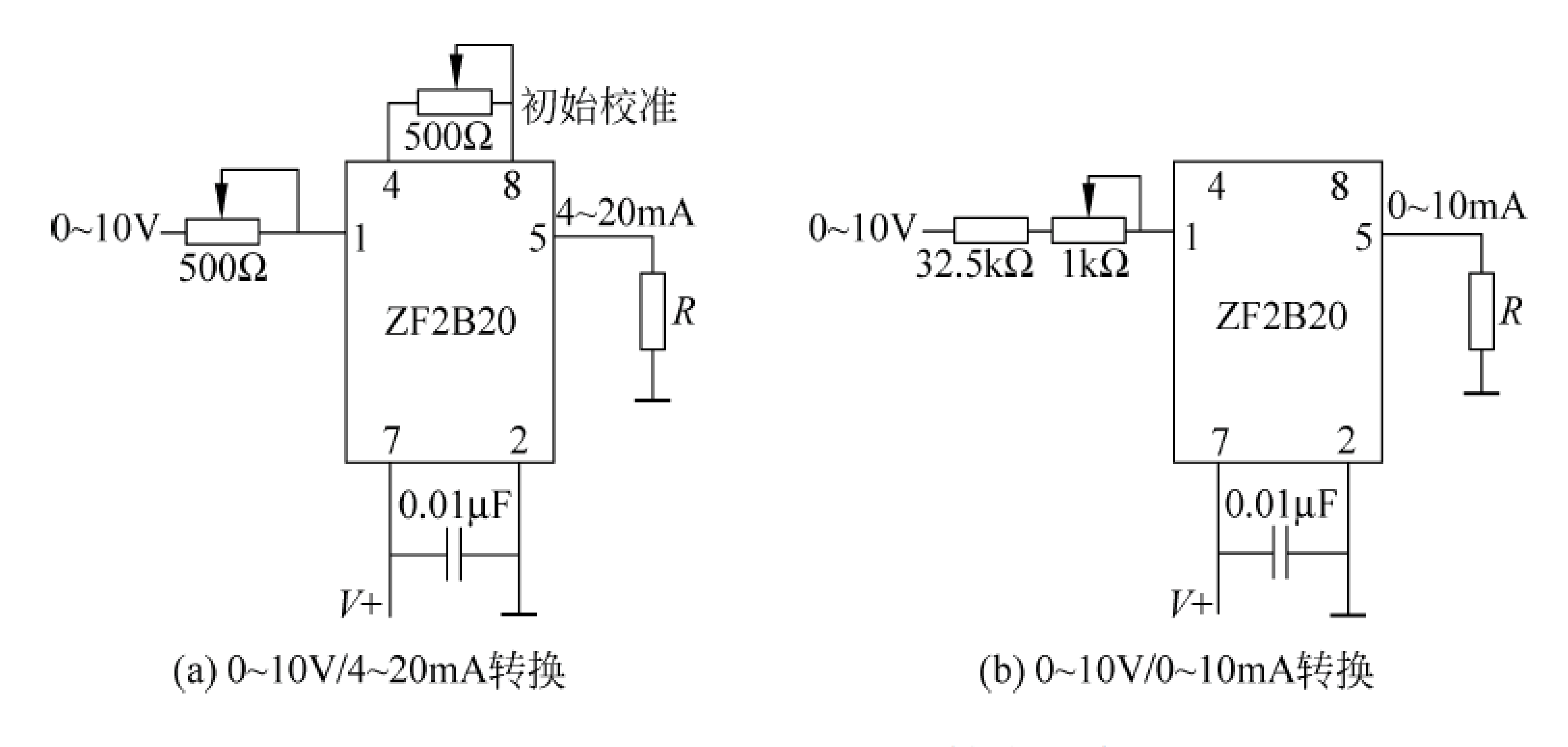


图 3.67 ZF2B20 的 V/I 转换电路

3.5　输入输出板卡举例

下面以研华公司的 PCI-1710HG 板卡为例，对输入输出板卡做一简单介绍。

3.5.1　PCI-1710HG 板卡简介

PCI-1710HG 是一款 PCI 总线的多功能数据采集卡，它包含五种最常用的测量和控制功能：12 位 A/D 转换、D/A 转换、数字量输入、数字量输出及计数器/定时器功能。具体特点如下。

- 16 路单端或 8 路差分模拟量输入，或组合方式输入。
- 12 位 A/D 转换器，采样速率可达 100kS/s(kilo Samples per second)。
- 每个输入通道的增益可编程。
- 单端或差分输入自由组合。
- 板载 4K 采样 FIFO 缓冲器。
- 两路 12 位模拟量输出。
- 16 路数字量输入和 16 路数字量输出。
- 可编程触发器/定时器。
- PCI 总线数据传输。

图 3.68 所示即为 PCI-1710HG 板卡及其配套产品，包括 PCL-10168 连接电缆和 ADAM-3968 接线端子。其主要功能有以下几个方面。

(1) 即插即用功能：PCI-1710HG 完全符合 PCI 规格 Rev2.1 标准，支持即插即用。在安装插卡时，用户不需要设置任何跳线和 DIP 拨码开关。所有与总线相关的配置，如基地址、中断，均由即插即用功能完成。

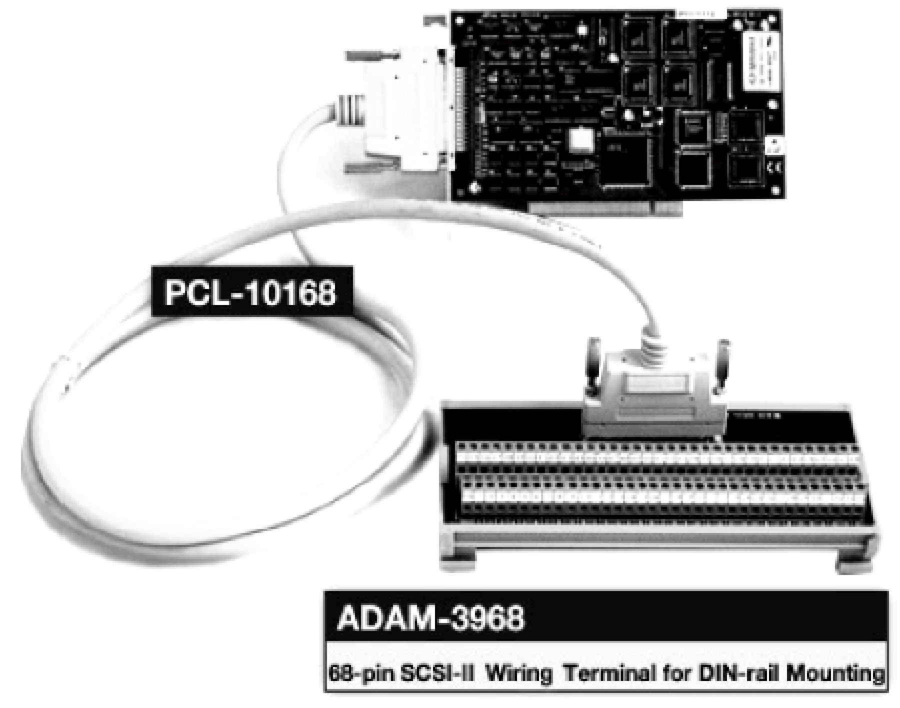


图 3.68　PCI-1710HG 板卡及其配套产品

(2) 单端或差分混合的模拟量输入功能：PCI-1710HG 有一个自动通道/增益扫描电路。该电路能代替软件控制采样期间多路开关的切换。卡上的 SRAM 存储了每个通道不同的增益值及配置。这种设计能对不同通道使用不同增益，并自由组合单端和差分输入来完成多通道的高速采样。

(3) FIFO(先入先出)存储器功能：卡上有一个 FIFO 缓冲器，它能存储 4K 的 A/D 采样值。当 FIFO 半满时，PCI-1710HG 会产生一个中断。

(4) 可编程计数器功能：PCI-1710HG 提供了可编程的计数器，用于为 A/D 变换提供可触发脉冲。计数器芯片为 82C54 或与其兼容的芯片，它包含了三个 16 位的 10MHz 时钟的计数器。其中有一个计数器作为事件计数器，用于对输入通道的事件进行计数。另外两个级联在一起，用作脉冲触发的 32 位定时器。

(5) 采用了降低噪声的特殊屏蔽电缆：如图 3.68 中配套 PCL-10168 屏蔽电缆是专门

为 PCI-1710/1710HG 所设计的,它用来降低模拟信号的输入噪声。该电缆采用双绞线,并且模拟信号线和数字信号线是分开屏蔽的。这样能使信号间的交叉干扰降到最小。

(6) 短路保护功能:PCI-1710/1710HG 在+12V(DC)/+5V(DC)输出管脚处提供了短路保护器件,当发生短路时,保护器件会自动断开,停止输出电流,直到短路被清除大约两分钟后,管脚才可开始电流输出。

3.5.2 板卡的基本应用

用 PCI-1710HG 板卡构成一个基本的计算机控制系统需首先进行板卡的安装,安装步骤如图 3.69 所示。

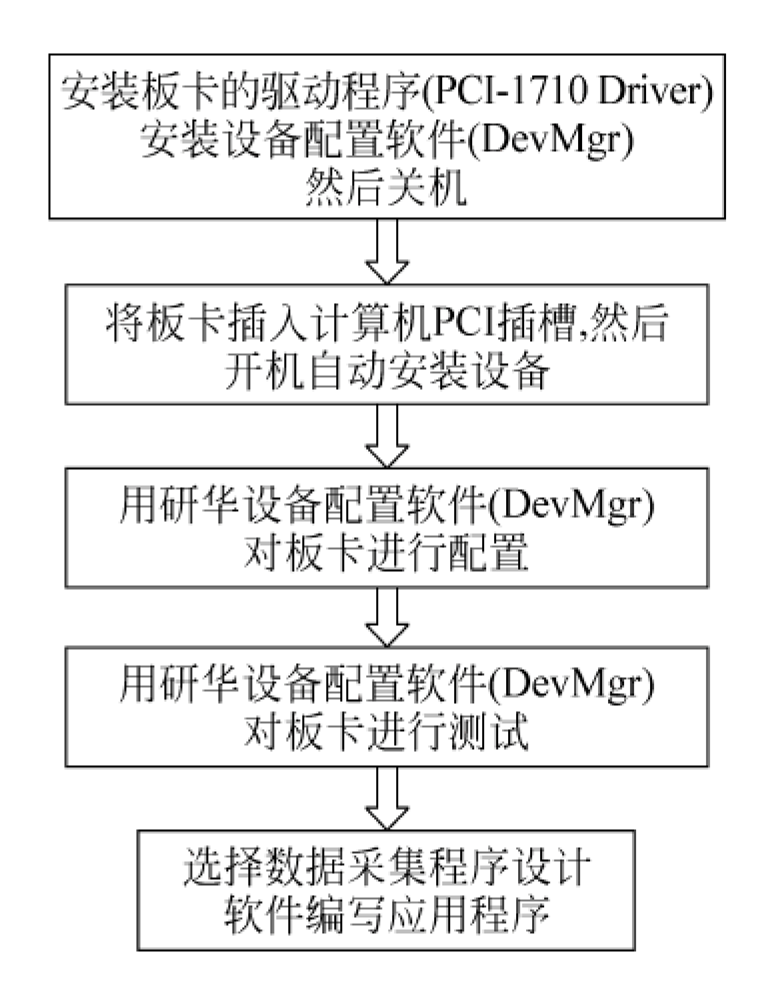


图 3.69 PCI-1710HG 板卡安装流程图

然后利用 PCL-10168 连接电缆将板卡与 ADAM-3968 接线端子板相连,如图 3.68 所示。这样,各种外部输入输出信号就可以通过接线端子与计算机通信了。构成系统的框图如图 3.70 所示。

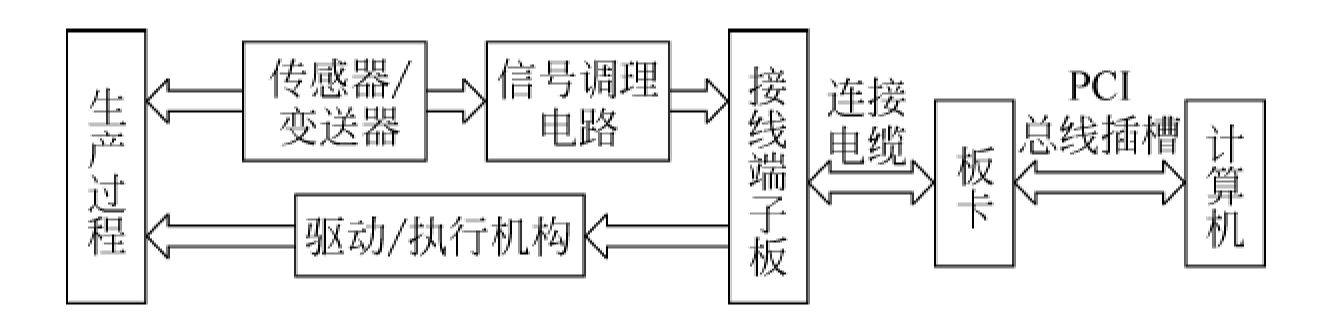


图 3.70 基于 PCI-1710HG 板卡的计算机控制系统框图

习题

1. 请说出计算机控制系统常用的主控制器有哪些?各有什么特点?
2. 总线可分为哪几类?分别用于什么场合?试各列举几个。
3. 计算机控制系统的开关量输出信号分别用于控制什么对象?试举例说明。
4. 试用 AEN(DMA 地址使用信号,高电平有效)信号、地址信号 $A_0 \sim A_9$、74HCT138

芯片，以及其他门电路，搭建一个地址译码电路，可寻址的地址分别为 3A0H～3A7H。

5. 试用 PCI-1710HG 总线工业板卡实现模拟量采集。画出接口电路原理图，并给出 A/D 转换程序。

6. 用 ADC0808 测量某罐温度，其温度波动范围 20～50℃，线性温度变送器输出 0～5V，试求测量该温度的分辨率和精度。

7. 数字量过程通道由哪些部分组成？各部分作用分别是什么？

8. 采用 74LS244 和 74LS273 与 PC 总线工业控制机接口，设计 8 路数字量(开关量)输入接口和 8 路数字量(开关量)输出接口，请画出接口电路原理图，并分别编写数字输入和数字输出程序。

9. 在数据采样系统中，是不是所有的输入通道都需要采样保持器？为什么？

第4章 硬件系统仿真技术

计算机仿真就是利用一些专用的仿真软件，建立被仿真系统的模型，对系统的结构、功能、属性、行为，以及进行系统控制的人的思维过程和方式、方法进行比较接近实际的动态模仿，以得到能够指导实践操作的理论依据；它具有便捷、高效、安全、受环境条件的约束较少、可改变时间比例尺等优点，已成为分析、设计、运行、评价、培训系统的重要工具。

硬件系统仿真软件种类繁多，但其功能会有所侧重，本章主要介绍怎样使用 Proteus 软件的 ISIS 设计环境进行计算机控制系统中硬件电路的设计与仿真。

4.1 Proteus 软件概述

Proteus 软件是英国 Labcenter Electronics 公司出版的 EDA 工具软件，它从原理图布图、代码调试到单片机与外围电路协同仿真，以及 PCB 设计等方面，真正实现了从概念到产品的完整设计，是一款进行控制系统设计与仿真的不可或缺的软件工具。它不仅可以为设计人员提供一个高效的设计与仿真的平台，更重要的是可以为初学者提供一个廉价、便捷的设计与验证的专业工具。

Proteus 软件具有两个主要设计环境：ISIS 和 ARES。ISIS 为智能原理图输入系统，是系统设计与仿真的基本平台；ARES 为高级 PCB 布线编辑软件，主要用于 PCB 的设计。本章主要介绍 ISIS 编辑环境，以及在此环境中怎样进行原理图设计与系统仿真。

Proteus ISIS 软件的主要特点有以下几方面。

(1) 具有强大的原理图绘制功能。Proteus 软件提供了三十多个元件库，数千种元件，元件涉及数字、模拟、交流和直流等，同时支持用户自己建立新的元件库。

(2) 实现了单片机仿真和 SPICE 电路仿真相结合。具有模拟电路仿真、数字电路仿真、单片机及其外围电路组成的系统的仿真等功能；可进行 RS-232 动态仿真；具有 I^2C 调试器、SPI 调试器，具有键盘和 LCD 仿真的功能；有各种虚拟仪器，如示波器、逻辑分析仪、信号发生器等；还可以针对步进电机、伺服电机、直流电机、无刷直流电机等控制对象进行仿真。

(3) 支持各主流单片机系统的仿真。目前支持的单片机类型有：68000 系列、8051 系列、AVR 系列、PIC12 系列、PIC16 系列、PIC18 系列、Z80 系列、HC11 系列以及多种外围芯片。

(4) 提供软件调试功能。在硬件仿真系统中具有全速、单步、设置断点等调试功能，同

时可以观察各个变量、寄存器等的当前状态，因此在该软件仿真系统中，也具有这些功能；同时支持第三方的软件编译和调试环境，如 Keil C51 μVision2 等软件。

4.2　Proteus ISIS 设计界面

Proteus ISIS 运行于 Windows 98/2000/XP 环境，一般的 PC 配置就能满足要求。将软件安装完成之后，根据图 4.1 选择："开始"→"程序"→Proteus 7 Professional→ISIS 7 Professional，出现如图 4.2 所示画面，即可进入 Proteus ISIS 集成环境。

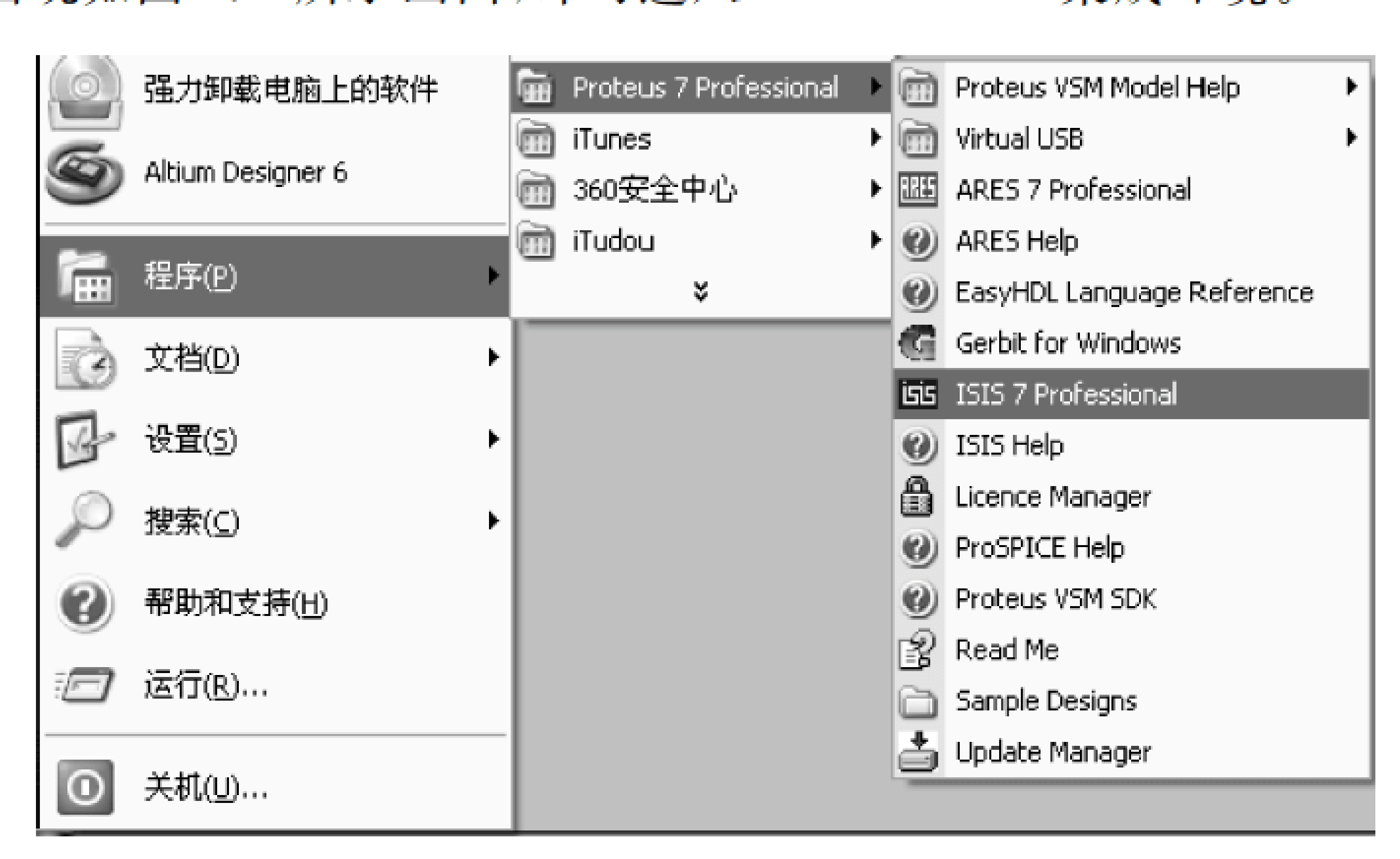


图 4.1　Proteus ISIS 打开步骤



图 4.2　Proteus ISIS 启动画面

4.2.1　基本界面介绍

Proteus ISIS 原理图设计界面及界面当中各部分内容如图 4.3 所示，主要包括：标题栏、主菜单、工具栏、绘图工具栏、状态栏、对象选择按钮、预览对象方位控制按钮、仿真进程控制按钮、预览窗口、对象选择器窗口、图形编辑窗口等。

点状的栅格区域为图形编辑窗口，左上方为预览窗口，左下方为元器件列表区，即对象选择窗口。图形编辑窗口用于放置元器件、连线、绘制原理图，并进行仿真。编辑原理图时，

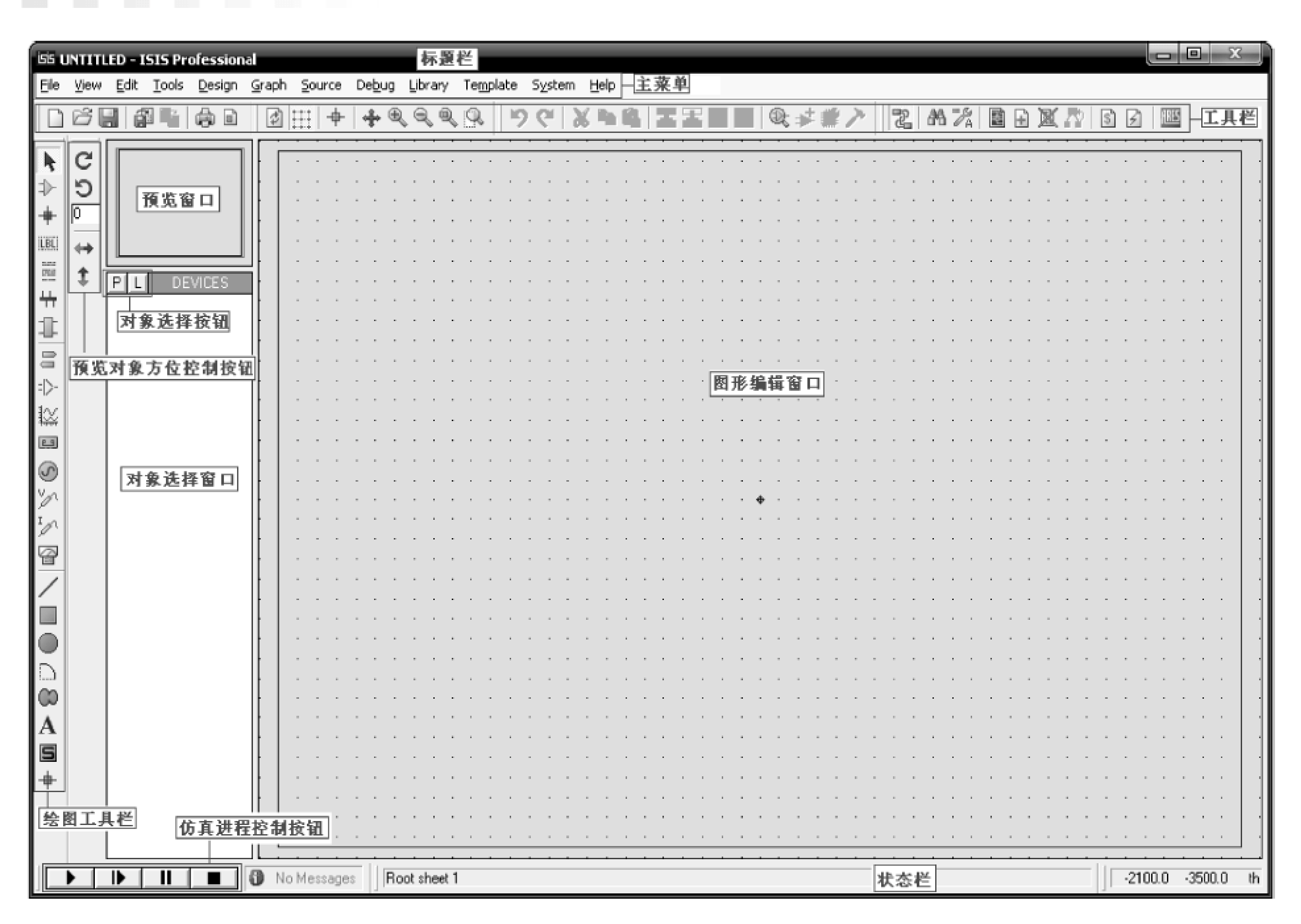


图 4.3 Proteus ISIS 的工作界面

预览窗口可以显示全部原理图,并在预览窗口中显示两个方框,蓝框表示当前页的边界,绿框表示当前编辑窗口显示的区域,在预览窗口上单击移动绿框,Proteus ISIS 将会以单击位置为中心刷新编辑窗口,如图 4.4 所示;当从对象选择窗口中选中一个新的元器件时,预览窗口可以预览选中的对象,可对选中的对象单击“预览对象方位控制”按钮进行翻转、镜像等操作。

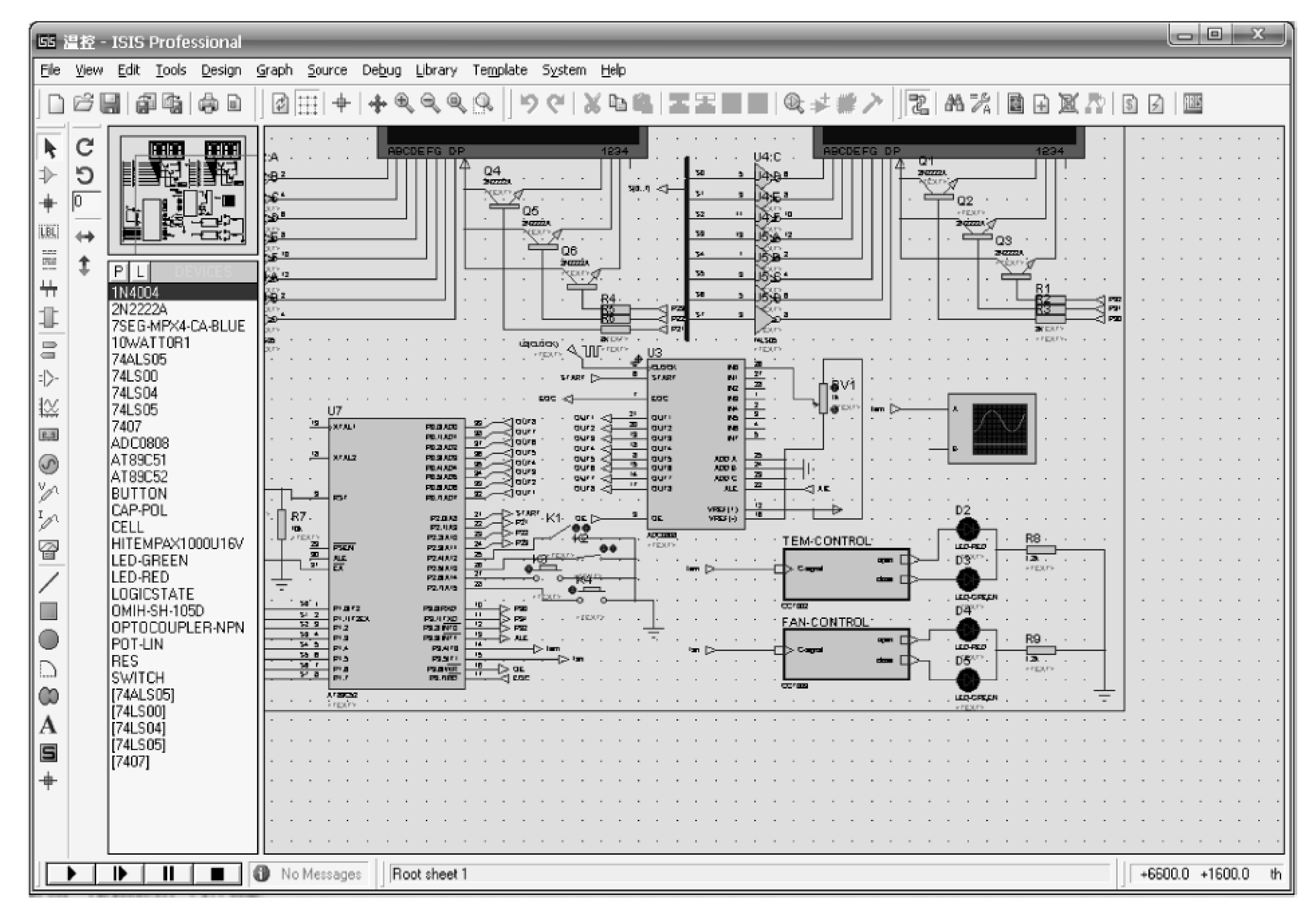


图 4.4 原理图设计时编辑界面

4.2.2 常用基本工具

常用基本工具主要指工具栏、绘图工具栏中所提供的各类按钮及图标，下面分别一一介绍。

工具栏中常用工具如表4.1所示，共38个，分别对应主菜单中相应功能，主要用于原理图文件的基本操作、设计环境的设定及原理图设计等。

表4.1　工具栏中常用工具

序号	图标	对应菜单	功能
1		File→New Design	新建设计
2		File→Open Design	打开设计
3		File→Save Design	保存设计
4		File→Import Section	导入部分文件
5		File→Export Section	导出部分文件
6		File→Print	打印
7		File→Set Area	设置区域
8		View→Redraw	刷新
9		View→Grid	栅格开关
10		View→Origin	原点
11		View→Pan	选择显示中心
12		View→Zoom In	放大
13		View→Zoom Out	缩小
14		View→Zoom All	显示全部
15		View→Zoom to Area	显示一个区域
16		Edit→Undo	撤销
17		Edit→Redo	恢复
18		Edit→Cut to clipboard	剪切
19		Edit→Copy to clipboard	复制
20		Edit→Paste from clipboard	粘贴
21		Block Copy	(块)复制
22		Block Move	(块)移动
23		Block Rotate	(块)旋转
24		Block Delete	(块)删除
25		Library→Pick Device/Symbol	拾取元器件或符号
26		Library→Make Device	制作元件
27		Library→Packaging Tool	封装工具
28		Library→Decompose	分解元器件
29		Tools→Wire Auto Router	自动布线器
30		Tools→Search and Tag	查找并标记
31		Tools→Property Assignment Tool	属性分配工具

续表

序号	图标	对应菜单	功能
32		Design→Design Explorer	设计资源管理器
33		Design→New Sheet	新建图纸
34		Design→Remove Sheet	移去图纸
35		Exit to Parent Sheet	转到主原理图
36		View BOM Report	查看元器件清单
37		Tools→Electrical Rule Check	生成电气规则检查报告
38		Tools→Netlist to ARES	创建网络表

绘图工具栏中常用工具如表 4.2 所示，共 15 个，主要用于变换模式、拾取元件、添加标注、放置导线、选择仿真工具、添加虚拟仪表等。

表 4.2 绘图工具栏中常用工具

序号	图标	名称	功能
1		Selection Mode	选择模式
2		Component Mode	拾取元器件
3		Junction Dot Mode	放置节点
4		Wire Lable Mode	标注网络标号
5		Text Script Mode	输入文本
6		Buses Mode	绘制总线
7		Subcircuit Mode	绘制子电路块
8		Terminals Mode	在对象选择器中列出各种终端
9		Device Pins Mode	在对象选择器中列出各种引脚
10		Graph Mode	在对象选择器中列出各种仿真分析的图表
11		Tape Recorder Mode	当对设计电路分割仿真时采用此模式
12		Generator Mode	在对象选择器中列出各种激励源
13		Voltage Probe Mode	可在原理图中添加电压探针
14		Current Probe Mode	可在原理图中添加电流探针
15		Virtual Instruments Mode	在对象选择器中列出各种虚拟仪器

除上述图标按钮外，系统还提供了 2D 图形模式图标按钮，如表 4.3 所示，主要包括画线、画矩形、画圆、画圆弧、画闭合线，添加文本、图像符号、标注等。这些图形对象可直接用于画图，例如用于创建新的库元件(元器件、符号、引脚和终端)，也可用于对所设计原理图进行一些必要的标注等。

表 4.3 2D 图形模式图标按钮

序号	图标	名称	功能
1		Line Mode	放置直线
2		Box Mode	放置矩形
3		Circle Mode	放置圆

续表

序号	图标	名　　称	功　　能
4		Arc Mode	放置圆弧
5		Closed Path Mode	放置闭合线
6		Text Mode	放置文字
7		Symbols Mode	放置图形符号
8		Markers Mode	放置图形标记

对于具有方向性的对象，系统还提供了各种旋转图标按钮，具体如表 4.4 所示。

表 4.4　旋转图标按钮

序号	图标	名　　称	功　　能
1		Rotate Clockwise	顺时针方向旋转按钮
2		Rotate Anti-Clockwise	逆时针方向旋转按钮
3		X-Mirror	水平镜像按钮
4		Y-Mirror	垂直镜像按钮

4.3 原理图编辑的基本操作

为了进行硬件系统的设计与仿真，首先需进行原理图图形文件的编辑。首先按照前面所述基本步骤进入如图 4.3 所示主界面。在 Proteus ISIS 窗口中选择 File→New Design 菜单项，弹出如图 4.5 所示创建新的设计文件对话框。选择合适的模板（通常选择 DEFAULT 模板），单击 OK 按钮，即可完成新设计文件的建立。当然，设计模板在原理图编辑的过程中还可以进行修改。

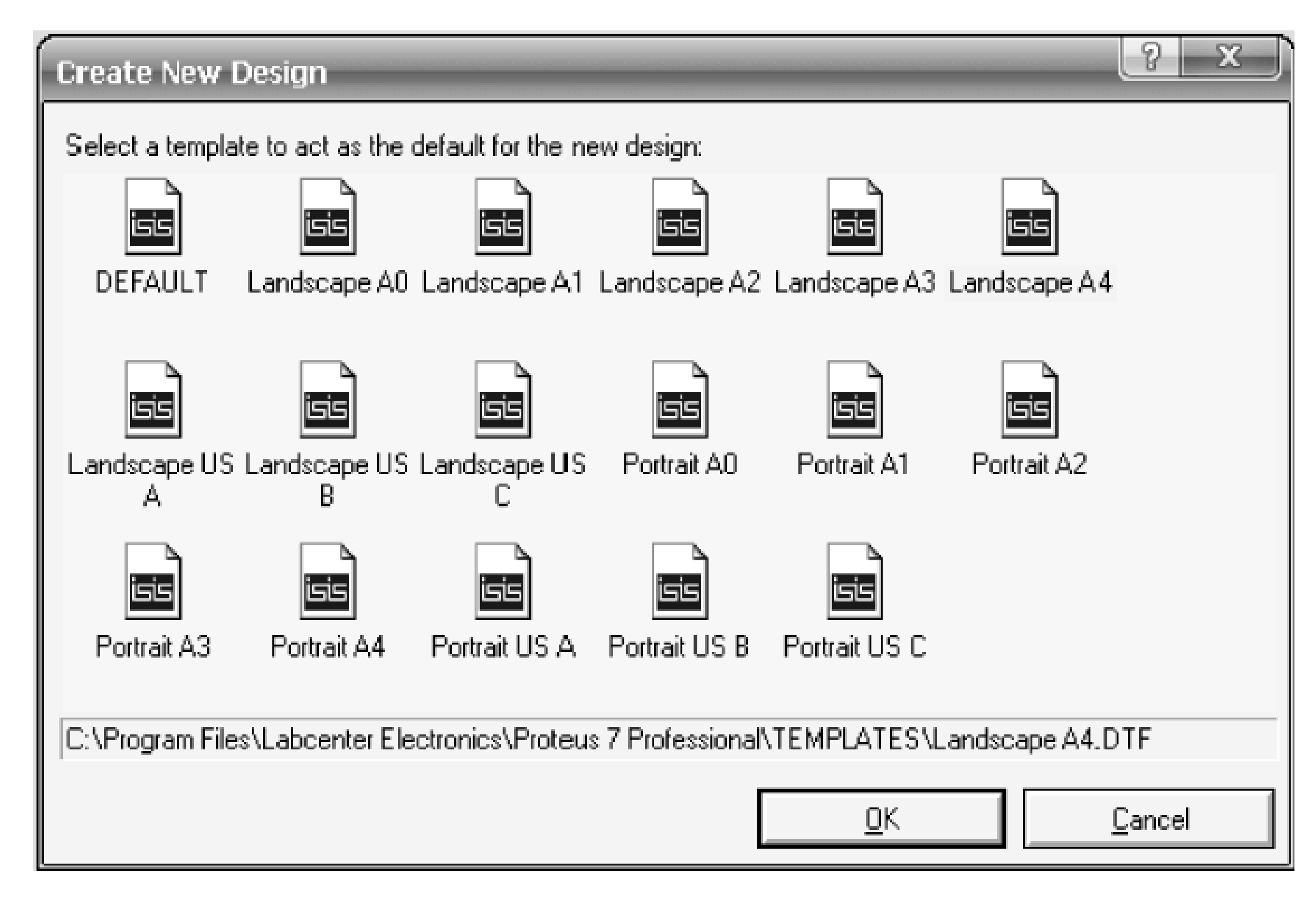


图 4.5　创建新的设计文件对话框

接着，选择 File→Save Design 菜单项，将弹出一个对话框，用于为设计文件命名，并选择保存路径。只需在“保存在”下拉列表框中选择目标存放路径，并在“文件名”框中输入设

计的文档名称。同时，保存文件的默认类型为 Design File，即文档自动加扩展名“. DSN”，单击“保存”按钮即可。

下面就可以进行原理图的编辑工作了，其基本操作主要包括元器件的选取、放置、删除、移动、复制、编辑，还有电路连线、添加标签等。

4.3.1 元器件的选取

首先用鼠标左键单击“对象选择按钮”（图 4.3 中）中的“P”，即可弹出拾取元件（Pick Devices）窗口，用于选取元件，如图 4.6 所示，该窗口主要显示内容为：关键词输入栏、元件库列表、元件库二级目录、元件库厂商列表、所选元件库中元件列表，以及所选元件原理图符号和 PCB 封装显示等。

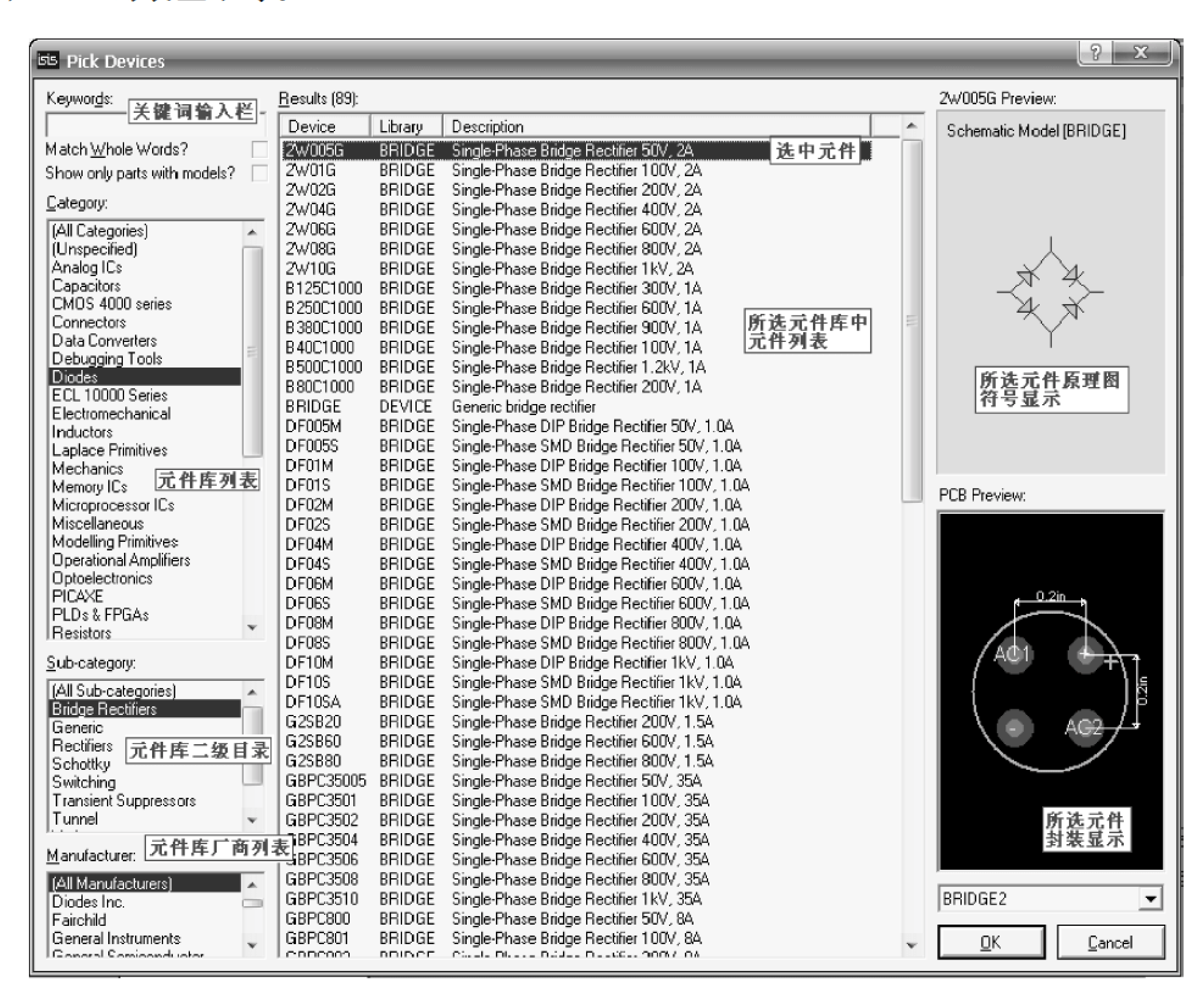


图 4.6 拾取元件（Pick Devices）窗口

选取元器件的方法有两种，一种是在关键字区域输入要查询的元器件名称，以缩小查找范围。如图 4.7 中在关键词输入栏输入“7404”，元器件列表区域便列出了所有与关键字相匹配的元器件，方便进行查找；找到后鼠标左键双击，或是单击后选 OK 按钮，即可拾取，于是，此元器件出现在图 4.3 所示的“对象选择窗口”，如图 4.8 所示。

另一种选取元件的方法如图 4.6 所示，在“元件库列表”中选择元件所属类，然后在“元件库二级目录”中选择所属子类，当对元器件的制造商有要求时，在“厂商列表”中选择厂商，即可在元件列表区域得到相应的元器件，然后进行选取。

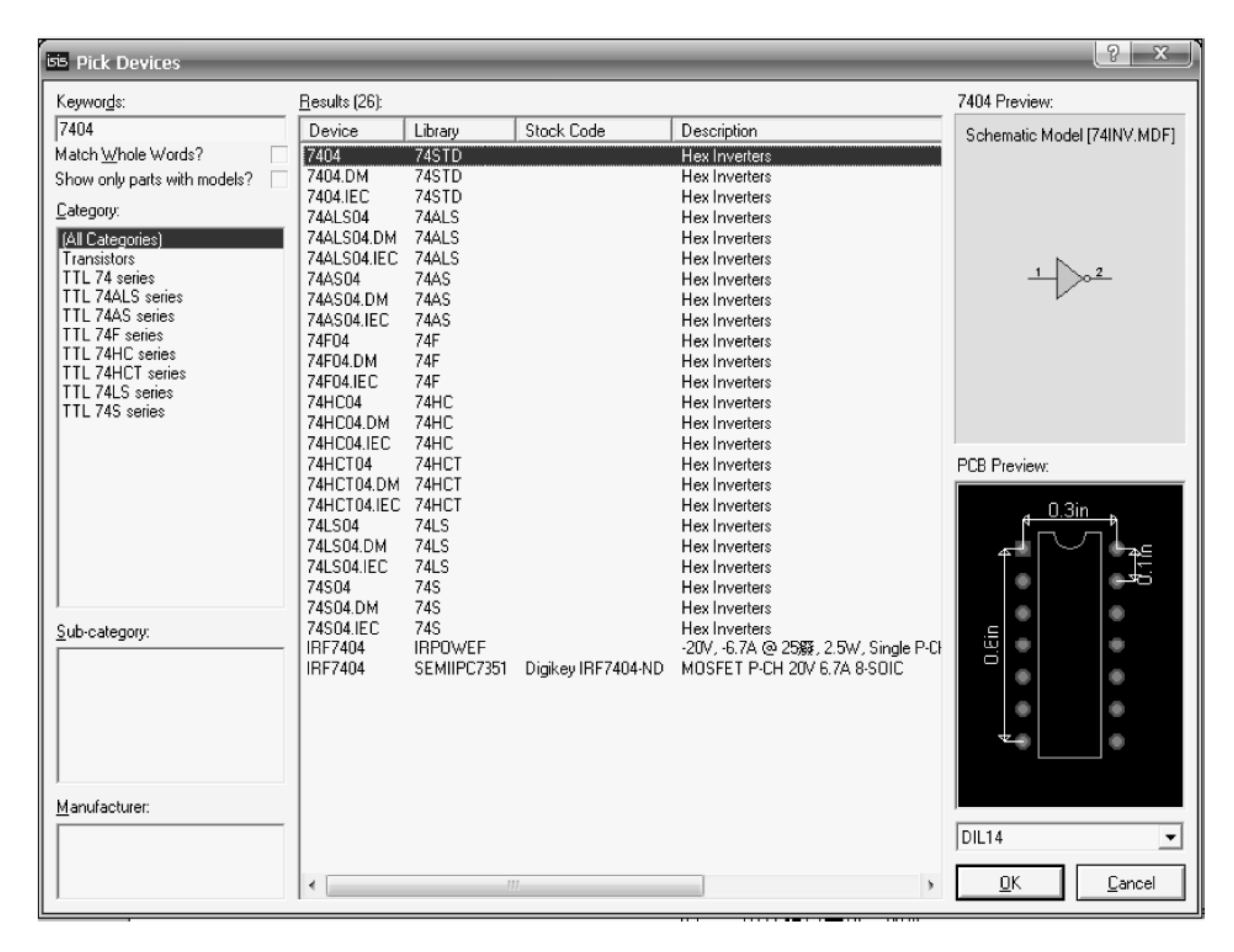


图 4.7　用关键词查找元器件

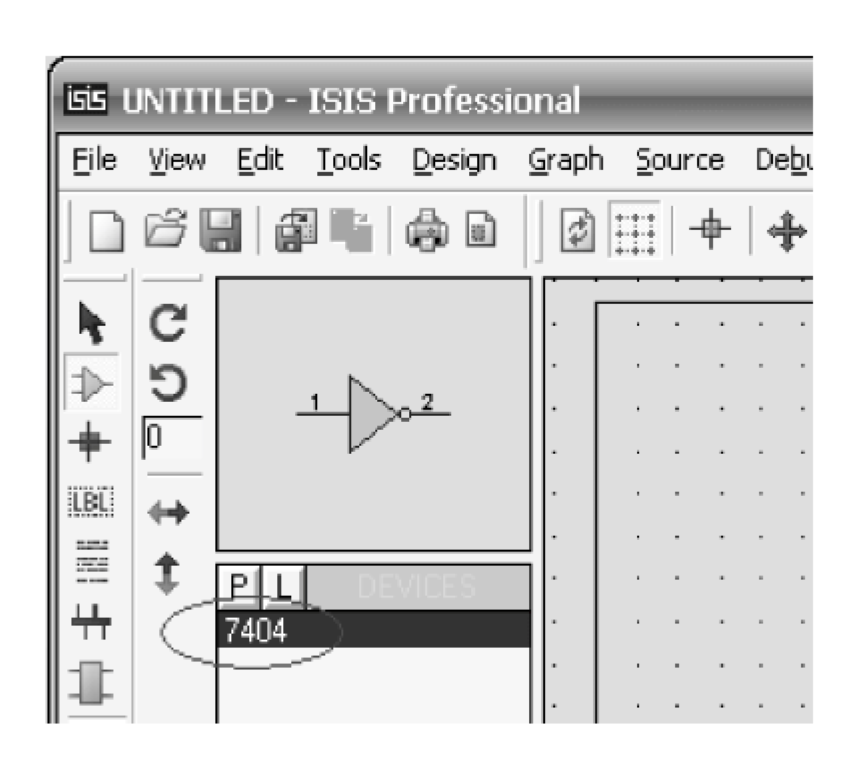


图 4.8　拾取元件显示

4.3.2　元器件的放置及编辑

当所需元件如图 4.8 那样被一一添加至“对象选择窗口”中后，就可以将元件放置于图形编辑窗口，以便进行原理图的设计了。就像图 4.4 左侧“对象选择窗口”中就列出了该设计所需的所有元器件。

在原理图中放置元器件的具体步骤如下。

(1) 单击“对象选择窗口”中的器件，在 Proteus ISIS 编辑环境主界面的“预览窗口”中即出现该器件的图标。

(2) 使用旋转图标按钮对元器件的方向进行调整。

(3) 在“图形编辑窗口”双击,元器件就被放置到原理图中。

重复上述步骤,分别将所需元器件放置到图形编辑窗口中,然后将光标指向编辑窗口的元器件,并单击该对象使其高亮显示,拖动该对象到合适的位置,也可再次进行旋转或镜像调整后,选择 View→Redraw 菜单项,刷新屏幕,便完成了元器件的放置,接下来就可以进行电路的连线了。

需删除元件或对象时,可用鼠标指向对象并双击鼠标右键,即可以删除该对象,同时删除该对象的所有连线。

当需拷贝一个对象或拷贝一整块电路时,用鼠标左键框选要复制的对象,被选择部分显示为红色,然后用鼠标左键单击工具栏中复制图标 ,把拷贝的轮廓拖到需要的位置,单击鼠标左键放置拷贝,重复此步骤可放置多个拷贝,单击鼠标右键结束。

对于一些元器件,如电容、电阻等,需要对元器件的参数进行修改时,可选择对元件进行编辑。以一个电阻元件为例,具体元件编辑方法如下。

(1) 通过元件拾取,将电阻元件放置于图形编辑窗口中。

(2) 双击元器件,弹出如图 4.9 所示对话框。

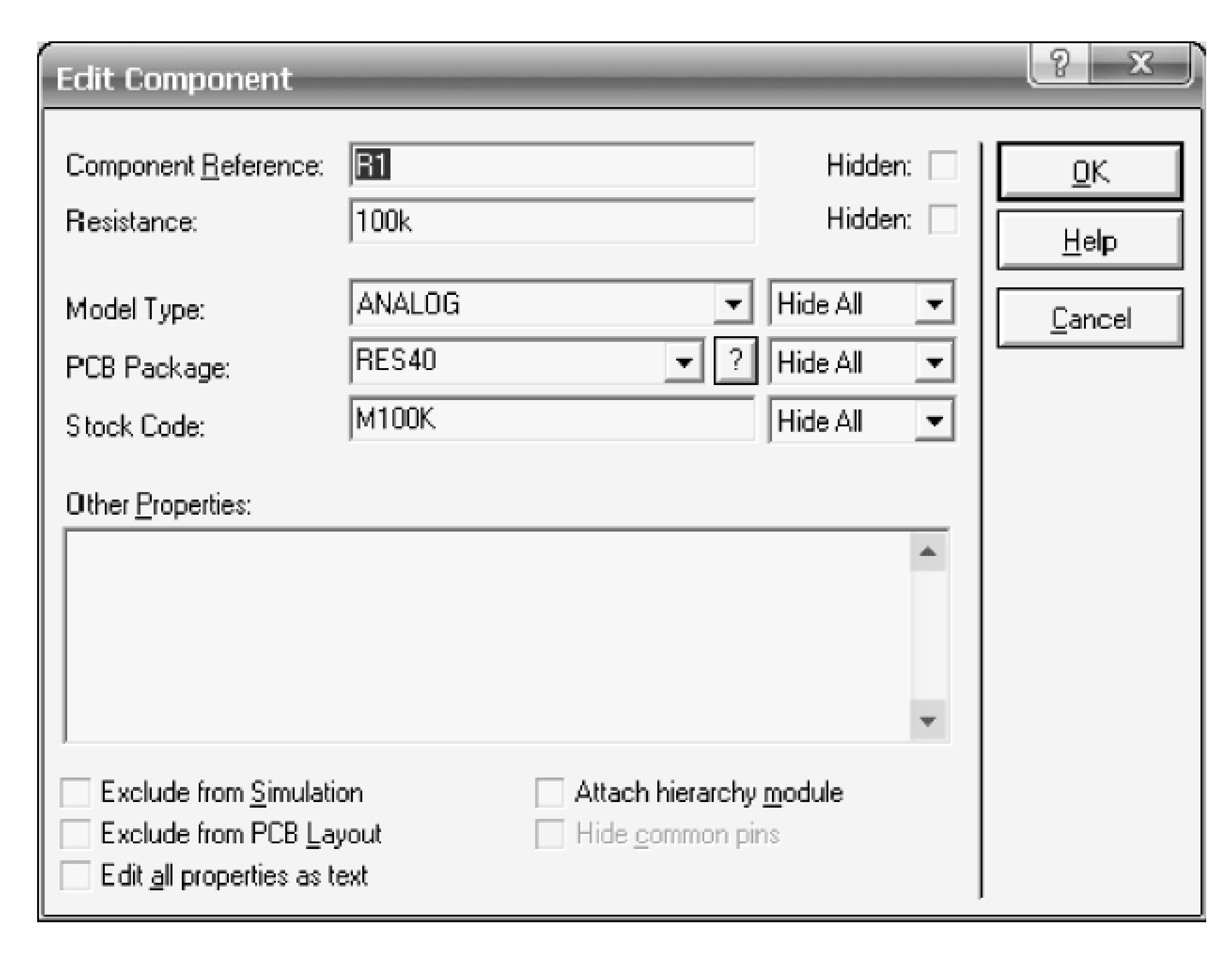


图 4.9 电阻元件编辑对话框

图 4.9 电阻元件编辑对话框中主要包含如下项目。

- Component Reference:元器件在原理图中的标号。
- Resistance:电阻阻值。
- Model Type:元件类型。
- Hidden:选择元器件参考号是否出现在原理图中。
- PCB Package:元器件的封装。
- Stock Code:库存代码。

(3) 对相关参数可进行修改,如阻值和标号等,然后单击 OK 按钮,结束元器件的编辑。

4.3.3　元器件的连线

对所有原理图所需元件进行拾取、放置和编辑之后，就可以进行元件的连线了。电路连线分为导线连接、添加网络标号连接、总线连接、使用输入/输出终端连接 4 种方式。

1. 导线连接

在两个对象间连线只需单击第一个对象连接点，然后，如果想让 ISIS 自动定出走线路径，只需再单击另一个连接点；如果想自己决定走线路径，只需在想要拐弯处单击即可。此外，当 3 根以上连线汇于一点时，ISIS 自动画出一个圆点，以示为各支路的公共交点。

2. 添加网络标号连接

当电路比较复杂，过多的连线会使电路中的导线相互交错，图纸显得凌乱，连接关系不容易辨别，可以利用标注网络标号来代替连接导线。具体方法是：单击绘图工具栏中图标 LBL，然后在需连接的端子或导线上添加网络标号，如图 4.10 中有三处添加了网络标号“tem”，则表示它们是短接在一起的。

3. 总线连接

在连接数据总线或其他平行的一组信号线时，可以将其连接成总线的方式。另外，Proteus ISIS 同时支持在层次模块间运行总线，也支持定义库元件为总线型引脚。在以总线方式进行连线时，需分为绘制总线、放置总线分支、添加网络标号三个步骤。

1）绘制总线

首先在绘图工具栏中单击总线(Buses Mode)图标，然后分别在期望总线起始端出现的位置单击鼠标左键，在总线路径的拐点处单击鼠标左键，在总线的终点单击鼠标左键，最后单击鼠标右键，即可结束总线放置。

2）放置总线分支

在 Protel 软件里，总线和总线分支是两个不同的命令。而在 Proteus 中，总线分支既可以用总线命令，也可以用一般连线命令。在使用总线命令画总线分支时，粗线自动变成细线。为了使电路图显得专业而美观，我们通常把总线分支画成与总线成 45°角的相互平行的斜线，下面举例来说明总线与总线分支的画法。

如图 4.11 所示，在 AT89C52 的 P1 口左侧先画一条自上而下的总线，确认主工具栏中的自动布线器为选中状态(为画斜线准备)。在 P1.0 引脚单击鼠标左键后松开，挪动鼠标画线，距总线一个背景栅格时，单击鼠标左键确认，然后左手按住 Ctrl 键不松，右手向右上移动鼠标，在与总线成 45°角相交时单击鼠标左键确认，即完成一条总线分支的绘制。下面需要画的几条不必再这样复杂，只需在分支的起始点双击即可完成。如画下一条 P1.1 引脚至总线的分支，把鼠标放置在 P1.1 引脚口位置，出现一个红色小方框，双击鼠标左键，自动完成像 P1.0 引脚到总线的一系列动作。依次完成所有总线分支的绘制。

需要指出的是，多条平行线在绘制时，均可采用以上简单的画法。

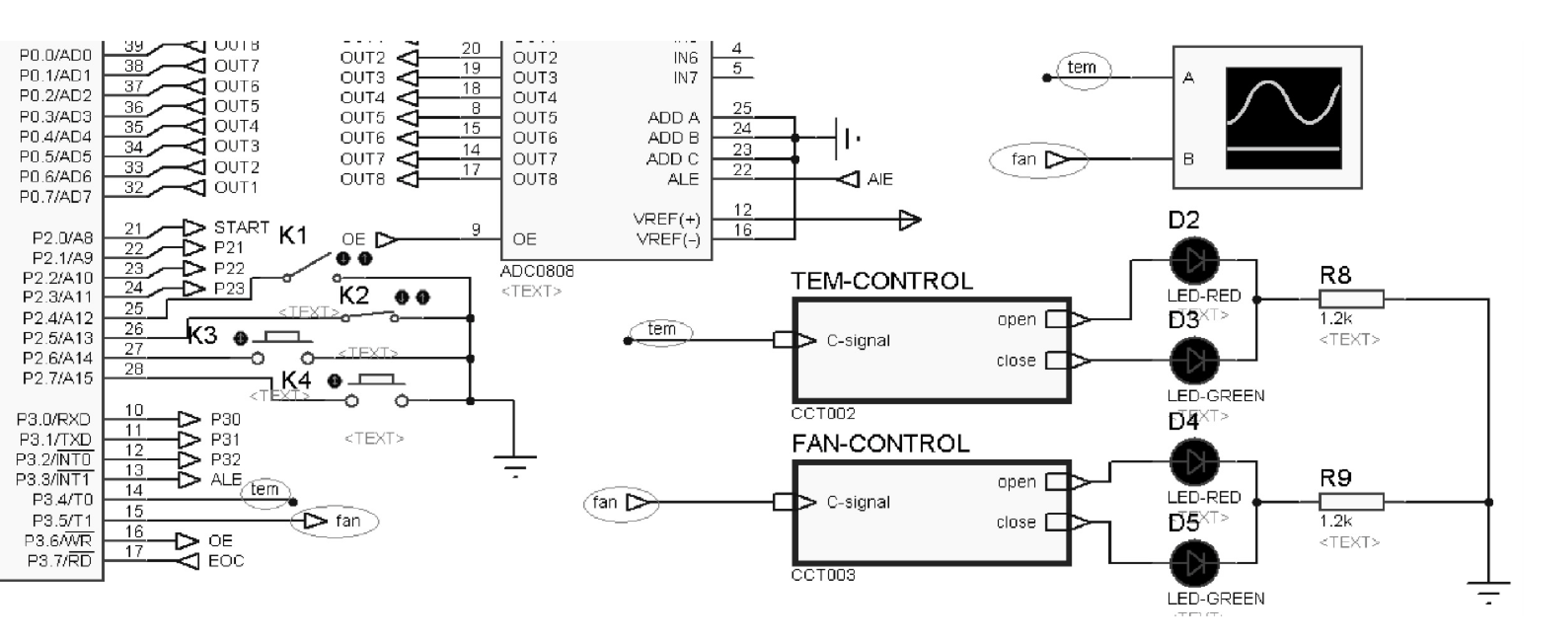


图 4.10 添加网络标号和连接终端的连接关系

3）添加网络标号

总线和总线分支画完之后，还需在每一条分支上添加网络标号，即从绘图工具箱中选择 Wire Label 图标 LBL 。把鼠标指向准备放置网络标号的总线分支位置，鼠标处出现一个“×”号，此时单击，出现 Edit Wire Label 对话框，在光标闪动位置填入标号名称即可。用总线方式连接 AT89C52 的 P1 口的电路如图 4.11 所示。

4. 添加终端连接

添加终端的连接方法类似网络标号的标注，可以在需要连接的两个接线端上分别连接输入终端和输出终端。具体方法是：用鼠标左键单击绘图工具栏中的 ，然后在对象选择窗口中选择 INPUT，使用输入终端；选择 OUTPUT，使用输出终端，如图 4.12 所示。以输出终端表示信号输出，以输入终端表示信号输入，并将这两个端子命名为同一名称。如图 4.10 中，有一个输出终端和两个输入终端都被命名为 fan，表明它们是相连的关系。

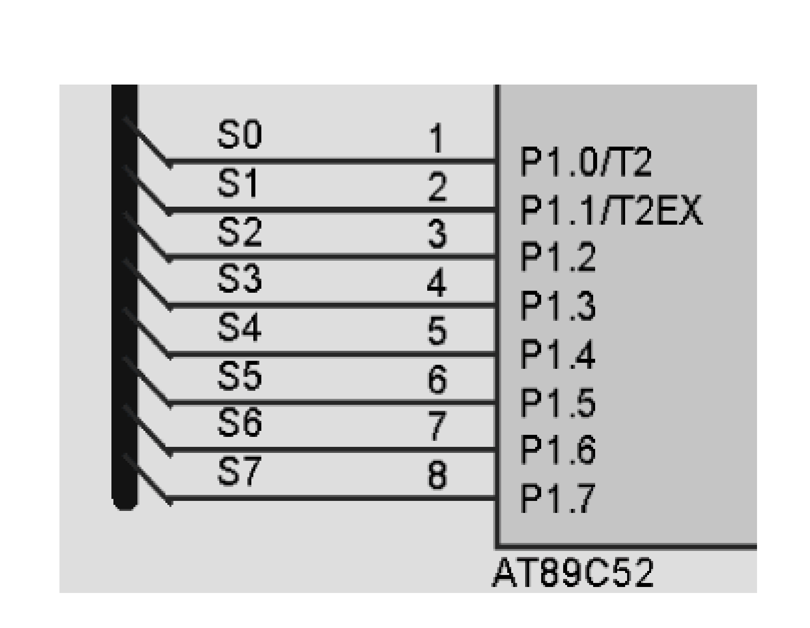


图 4.11　总线连接

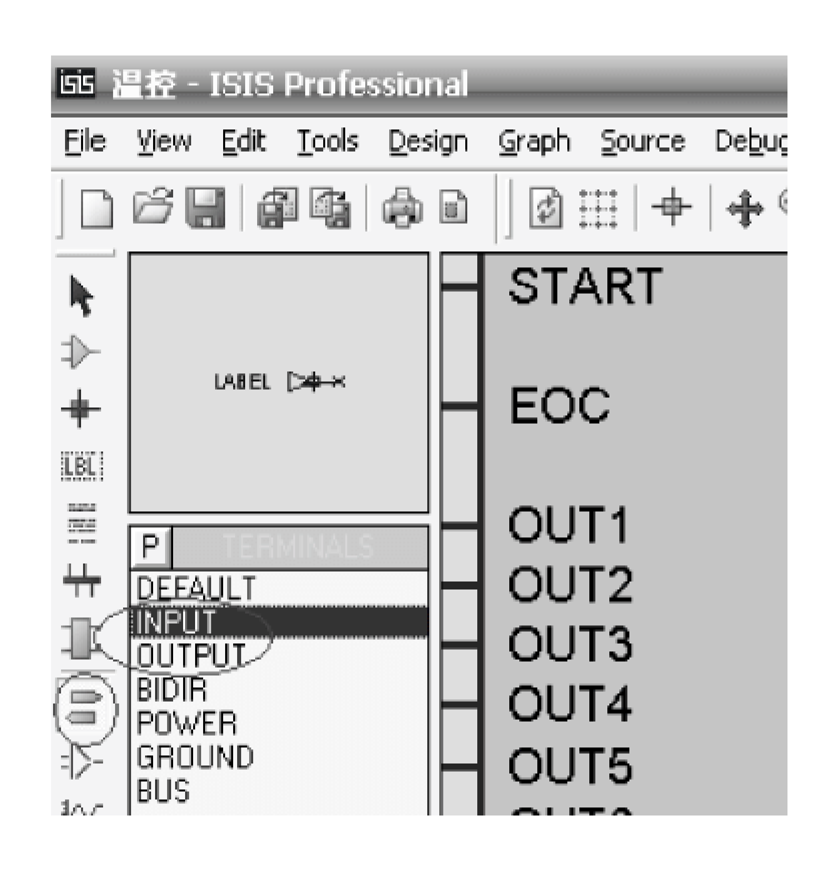


图 4.12　输入终端和输出终端的选取

4.3.4　编辑环境的设置

Proteus ISIS 编辑环境的设置主要包括图纸的选择、图纸的设置、文本编辑器的设置、栅格的设置等。如设置纸张的型号、标注的字体，是否显示图纸的栅格，以及栅格的大小等。

1. 图纸的设置

原理图的图纸不宜太大，当然也不宜太小，可在创建新的设计文件时进行图纸大小的设定，如图 4.5 所示。在编辑原理图的过程中，如果感觉图纸的尺寸不合适，也可随时进行修改，具体方法是：在 Proteus ISIS 主界面选择 System→Set Sheet Sizes 菜单项，将出现如图 4.13 所示对话框。在该对话框中用户可选择图纸的大小或自定义图纸的大小。

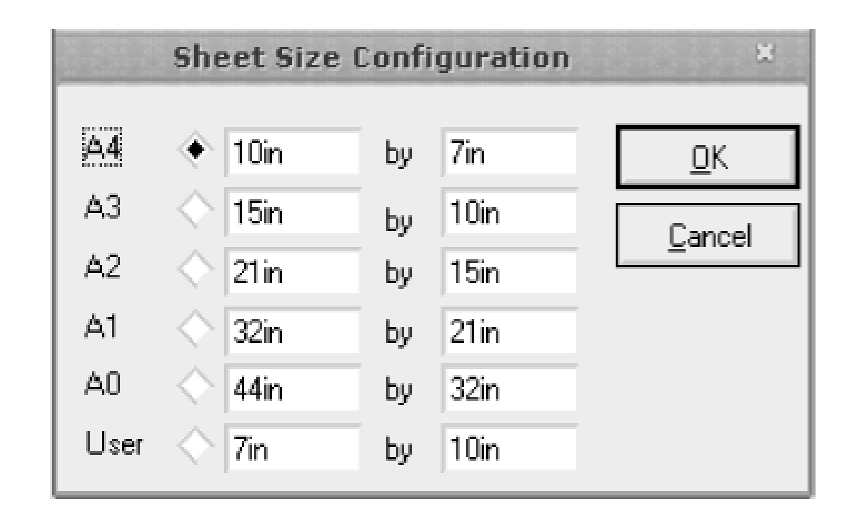


图 4.13　图纸大小设置对话框

2. 设置文本编辑器

如需设置文本格式，可在 Proteus ISIS 主界面中

选择 System→Set Text Editor 菜单项，弹出如图 4.14 所示对话框。在该对话框中可以对文本的字体、字形、大小、效果和颜色等进行设置。



图 4.14　设置文本格式

3. 设置栅格

在进行原理图设计时，图形编辑区中的栅格方便元器件的布局和对齐，也利于电路中导线的排列，但一个栅格的宽度决定了光标移动的单位移动量，例如画导线或放置元器件时，如果需要移动的位置小于当前栅格的尺寸，可对栅格的大小进行修改。具体方法是：选择 View→Snap 10th 菜单项，或 Snap 50th、Snap 0.1in、Snap 0.5in 项，可将栅格的尺寸调整为 10th、50th、0.1in 或 0.5in，系统默认值为 0.1in，如图 4.15 所示。

当需要对所设计的结果进行截图时，为了使图纸清晰，可将栅格隐去，不再显示，具体方法是：在主界面中选择 View→Grid 菜单项，或直接用鼠标单击工具栏中图标 ，即可将栅格隐去，当再次进行同样的操作时，又可将栅格显示出来。

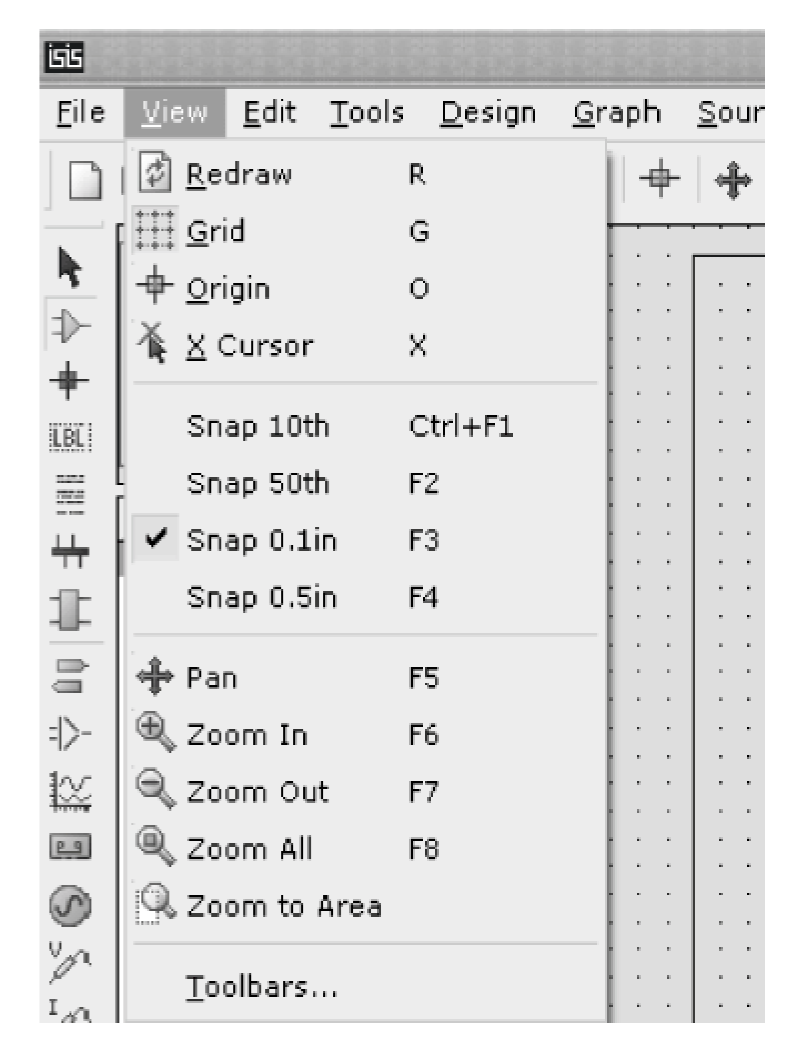


图 4.15　栅格尺寸修改

4. 动画选项设置

选择 System→Set Animation Options 菜单项，即可打开仿真电路设置对话框，如图 4.16 所示。

在该对话框中可以设置仿真速度、电压/电流的范围，同时还可对以下动态仿真选项进行设置。

- Show Voltage & Current on Probes：是否在探测点显示电压值与电流值。
- Show Logic State of Pins：是否显示引脚的逻辑状态。
- Show Wire Voltage by Colour：是否用不同颜色表示导线的电压。

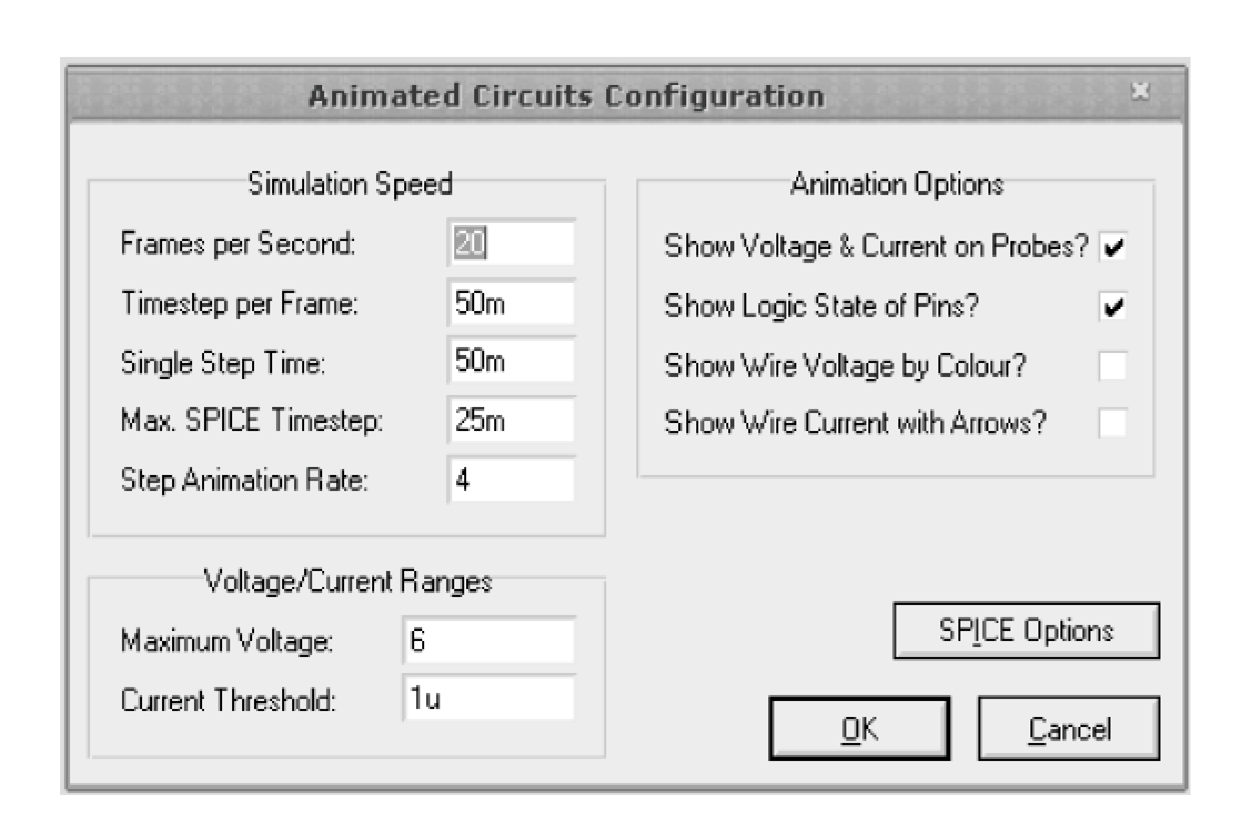


图 4.16　仿真电路设置对话框

• Show Wire Current with Arrows：是否用箭头表示导线的电流方向。

此外，单击 SPICE Options 按钮，弹出如图 4.17 所示交互式仿真选项对话框。在该对话框中还可通过选择不同的选项来进一步对仿真电路进行设置。

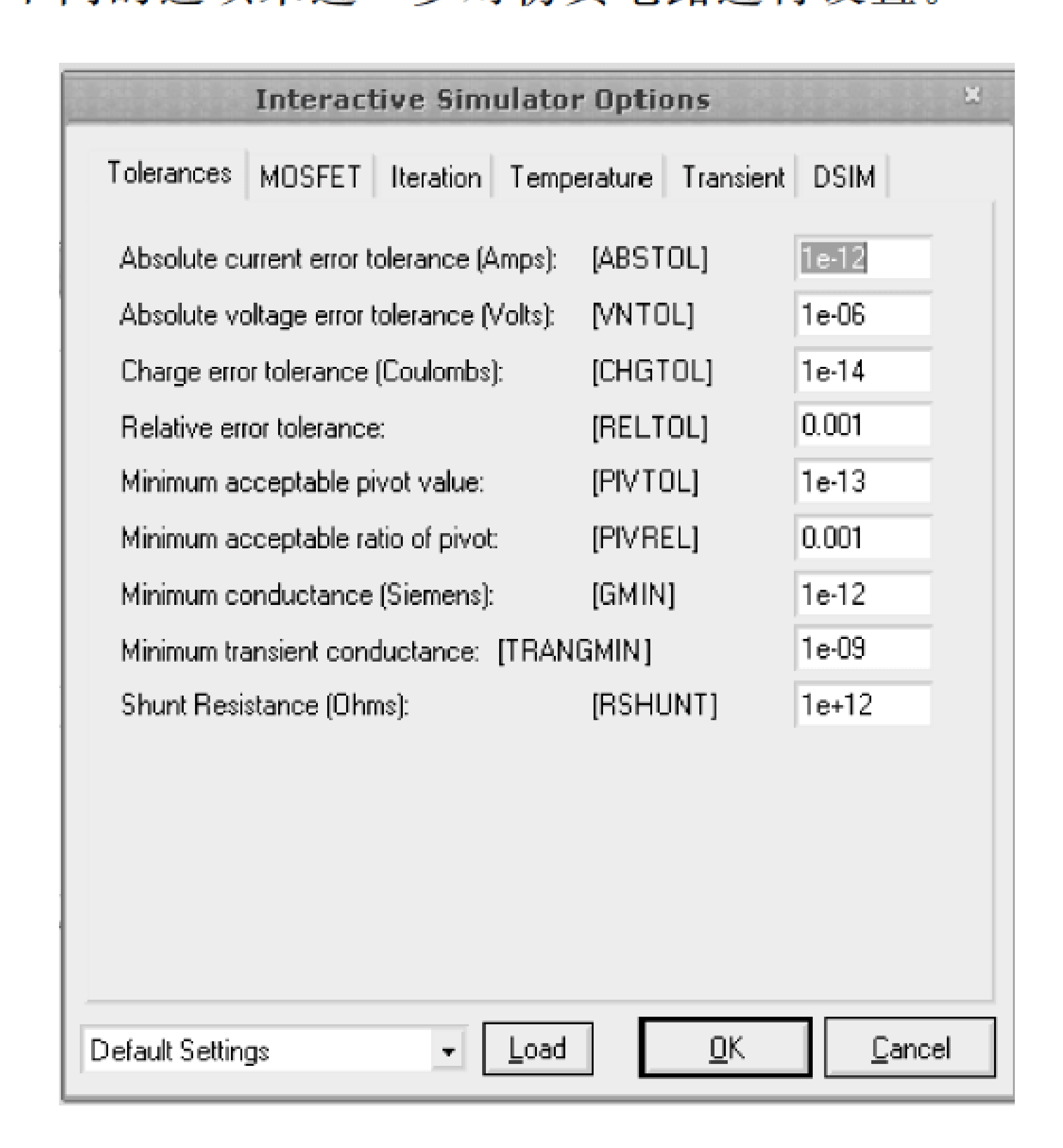


图 4.17　交互式仿真选项对话框

4.3.5　层次原理图设计

和多图纸设计过程一样，ISIS 支持层次电路设计。对于一个较大、较复杂的电路图，不可能一次完成，也不可能将这个电路图画在一张图纸上，更不可能由一个人单独来完成。利用层次电路图可以大大提高设计速度，也就是将这种复杂的电路图根据功能划分为几个模块，由不同的人员分别来完成各个模块，做到多层次并行设计。本节将通过一个具体的例子来介绍层次电路图的基本概念和绘制层次原理图的步骤与技巧。

如图 4.18 是一个层次电路（主电路），其中 TEM-CONTROL 和 FAN-CONTROL 为子电路，子电路的内部电路图如图 4.19 所示。

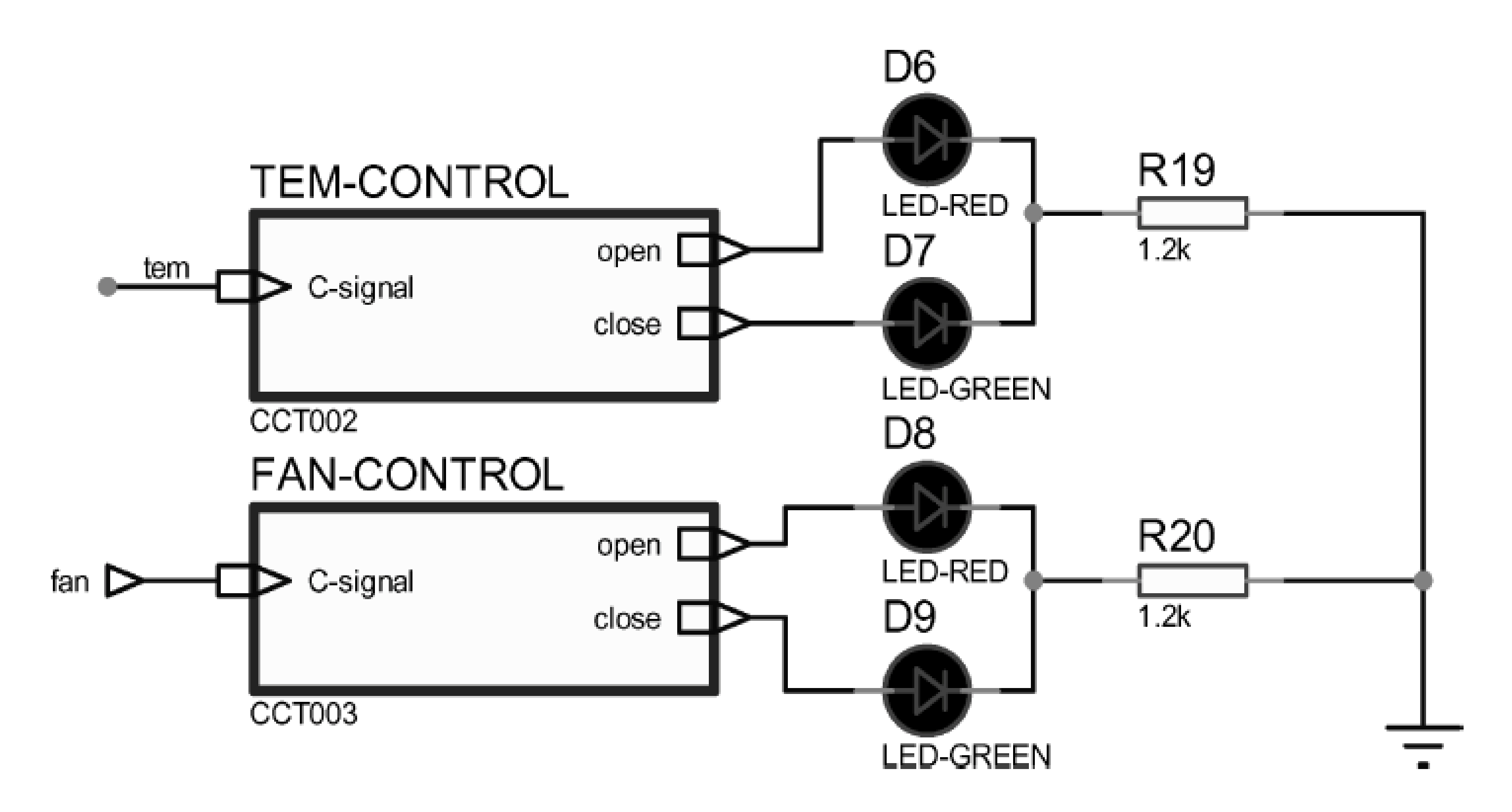


图 4.18　层次电路设计例图

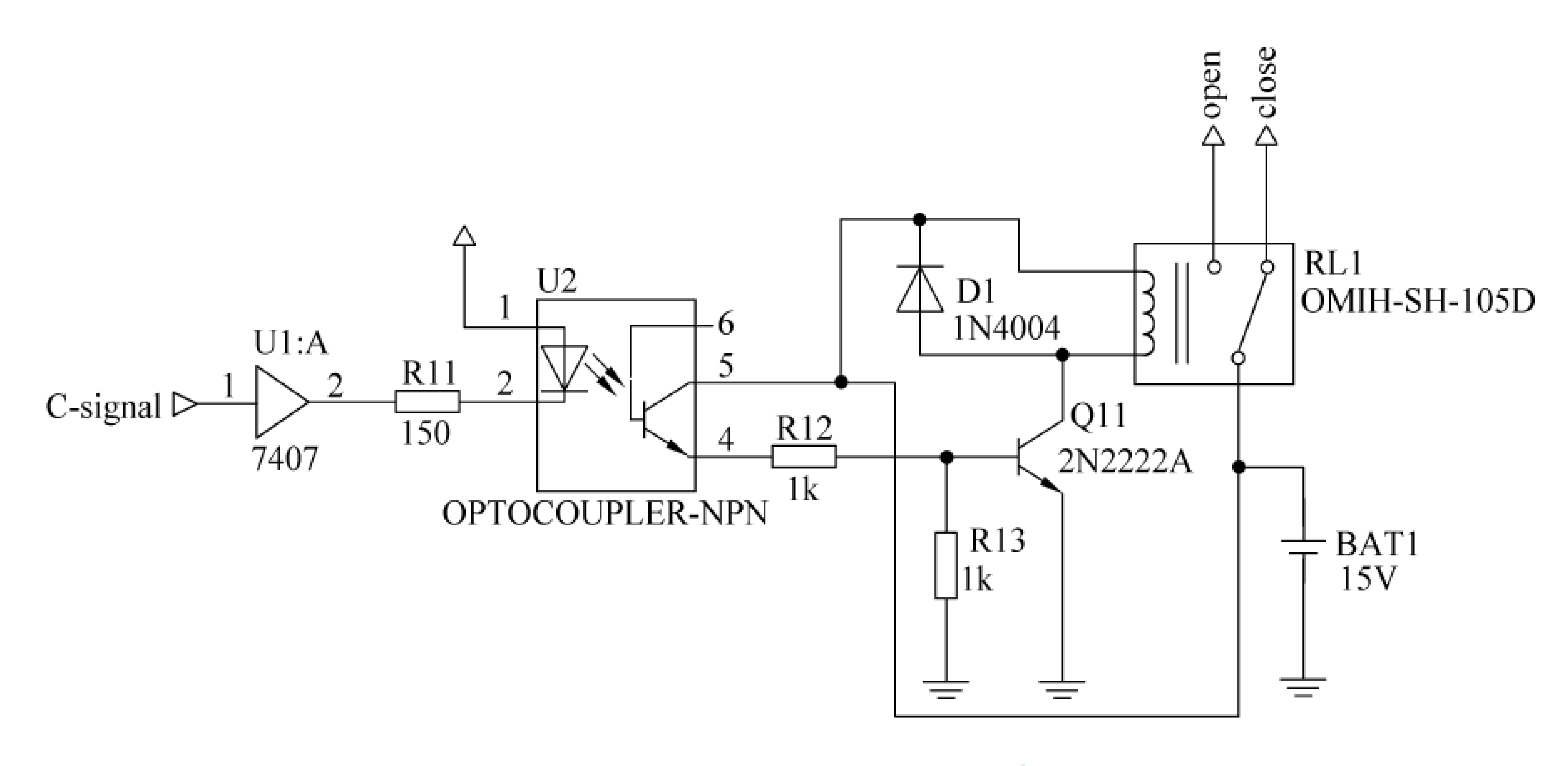


图 4.19　TEM-CONTROL 电路图

层次电路的设计步骤介绍如下。

1）画子电路模块

单击工具栏中的子电路工具，并在编辑窗口拖动，拖出子电路模块，如图 4.20 所示。从对象选择窗口中选择输入、输出端口，放置在子电路图的左侧和右侧。端口用来连接子电路和主电路。一般输入端口放在电路图模块的左侧，而输出端口放在右侧，如图 4.21 所示。

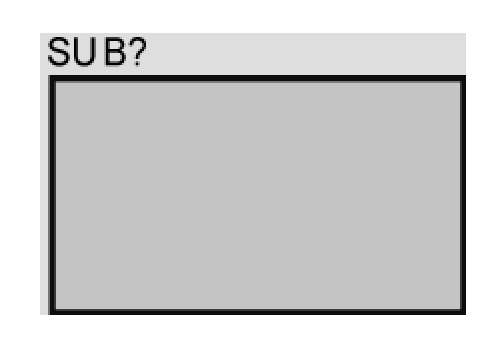


图 4.20　子电路图模块

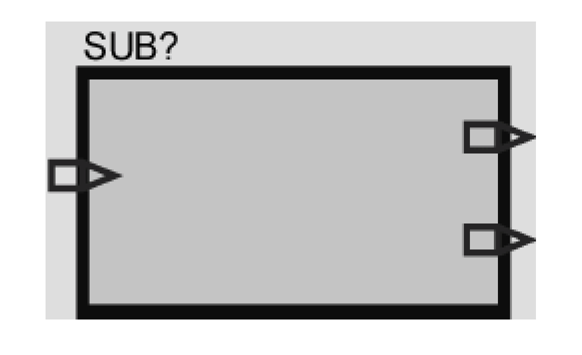


图 4.21　添加子电路图端口

2）编辑子电路及其端口名称

直接使用端口编辑对话框编辑端口名称，端口的名称必须与子电路的逻辑终端名称一致。例如，将光标放在输入端口上右击，在下拉窗口中选择 Edit Properties，然后输入端口

名称即可，如图 4.22 所示。本电路输入端口是 C-signal，输出端口是 open 和 close。

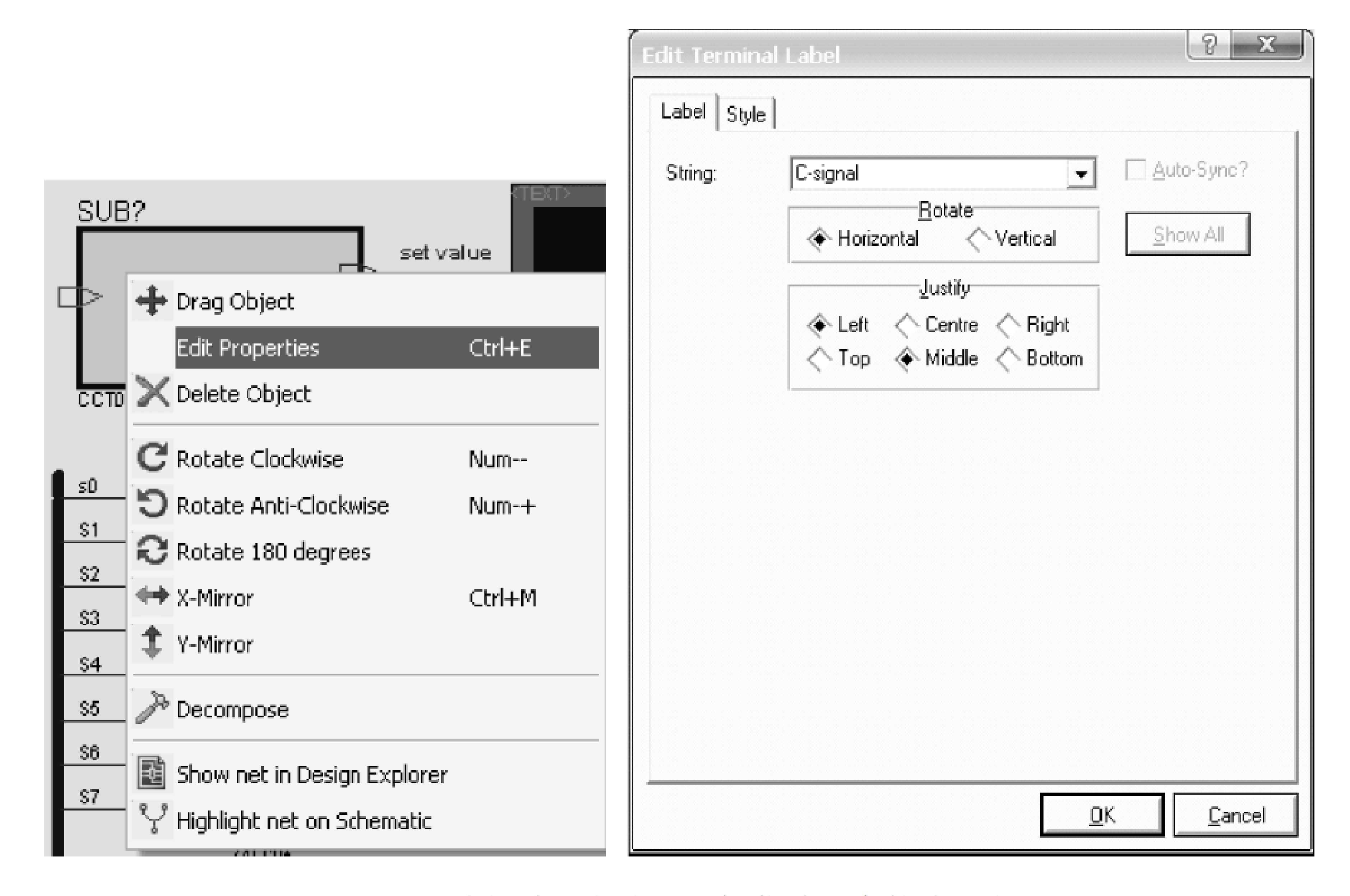


图 4.22　编辑端口名称的下拉菜单及参数输入窗口

同样，光标放在“SUB?”上，单击鼠标右键，选择 Edit Label，输入子电路名称，如图 4.23 所示。或者选中整个子电路模块，单击鼠标右键，选择 Edit Properties，如图 4.24 所示，即可打开子电路模块属性编辑对话框，如图 4.25 所示，子图框的 Name 输入“TEM-CONTROL”(实体名称)即可。在同一个图页，每个子电路必须有唯一的子图框名称，如 TEM-CONTROL 和 FAN-CONTROL。这时，子电路图模块如图 4.26 所示。当端口是低电平有效，输入的名称像 $\bar{R}$ 时，只需输入“$R”即可。

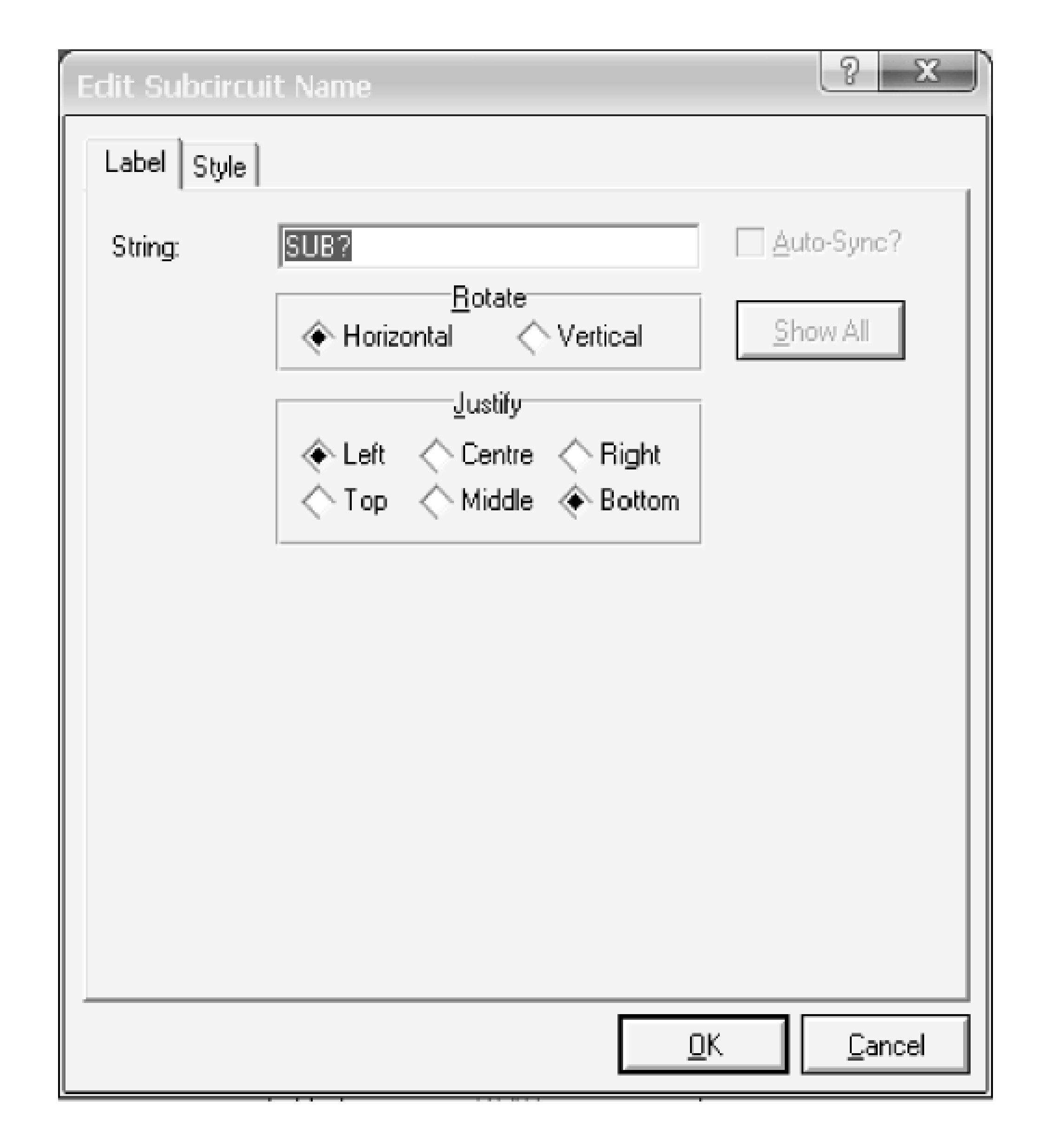


图 4.23　子电路图名称编辑窗口

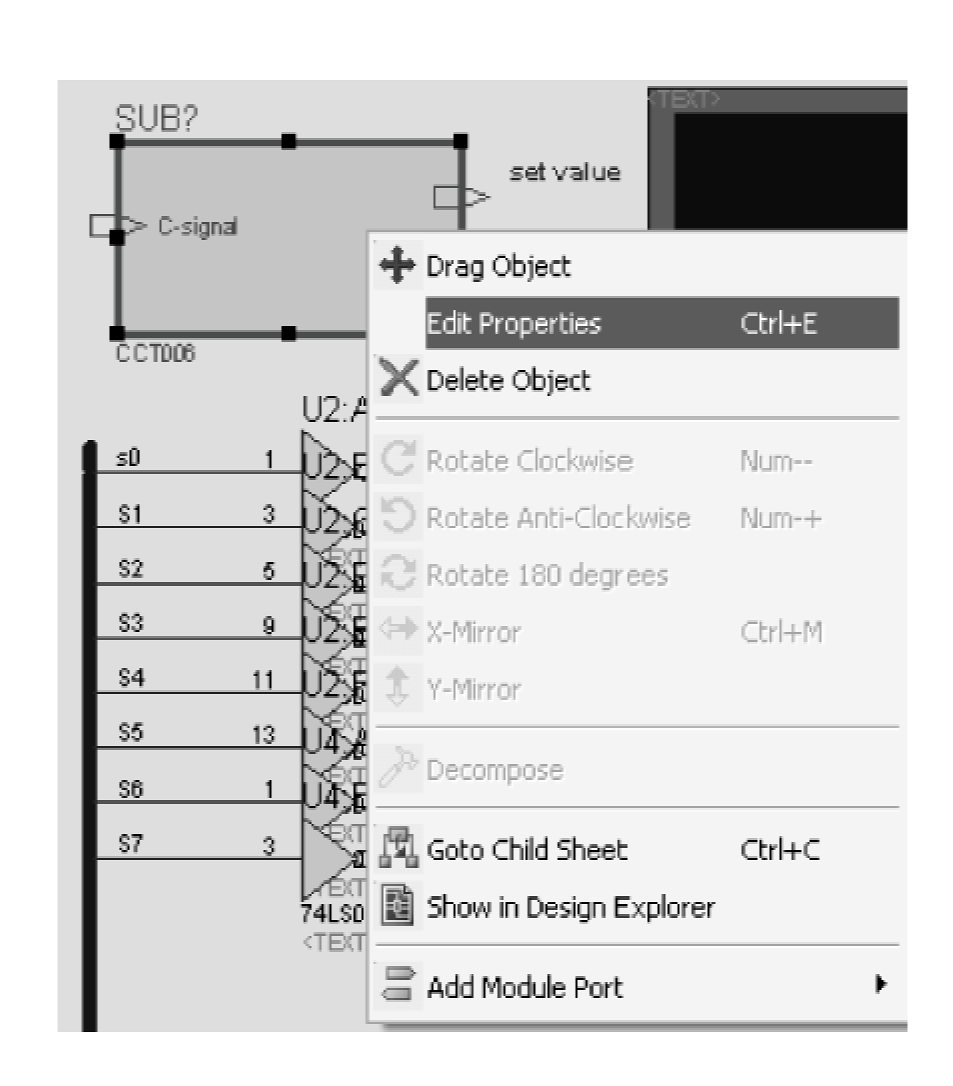


图 4.24　打开子电路模块属性编辑窗口

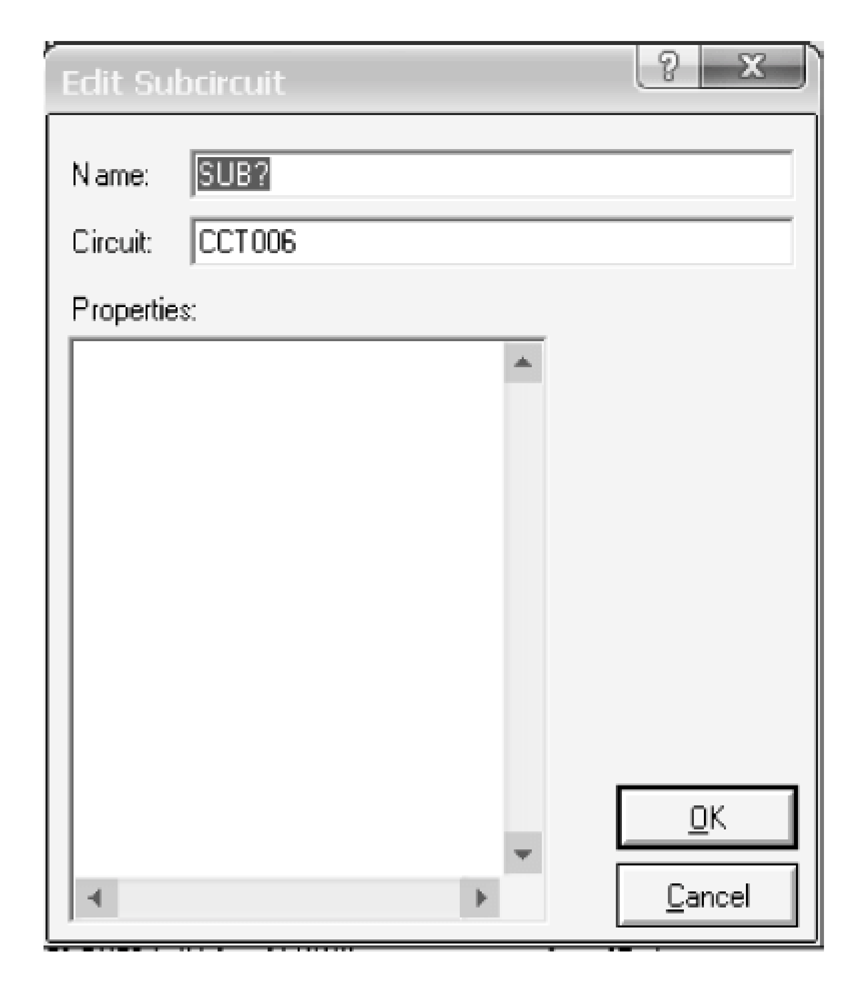


图 4.25　子电路模块属性编辑对话框

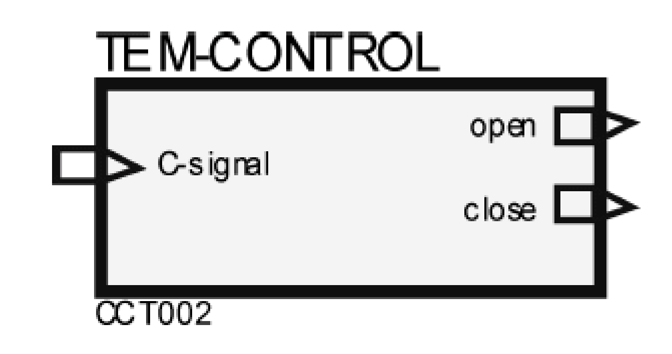


图 4.26　子电路图模块

3）子电路编辑

将光标放置在子电路模块上，单击鼠标右键，并选择菜单命令 Goto Child Sheet（默认组合键为 Ctrl+C），这时 ISIS 将加载一空白的子图页，操作如图 4.27 所示。在打开的空白子图页中可进行原理子图的编辑，结果如图 4.19 所示。单击工具箱中的按钮，可根据需要选择相应对象。需要电源时，选中对象选择窗口中的 POWER，则在预览窗口出现电源信号的图标，在原理图中单击，可在原理图中添加电源符号，选中电源符号，拖到合适的位置，并将其连接到电路。也可选中电源符号单击，进入电源编辑对话框，在 String 栏中分别输入 +15V、−15V，然后单击 OK 按钮，完成电源的放置。

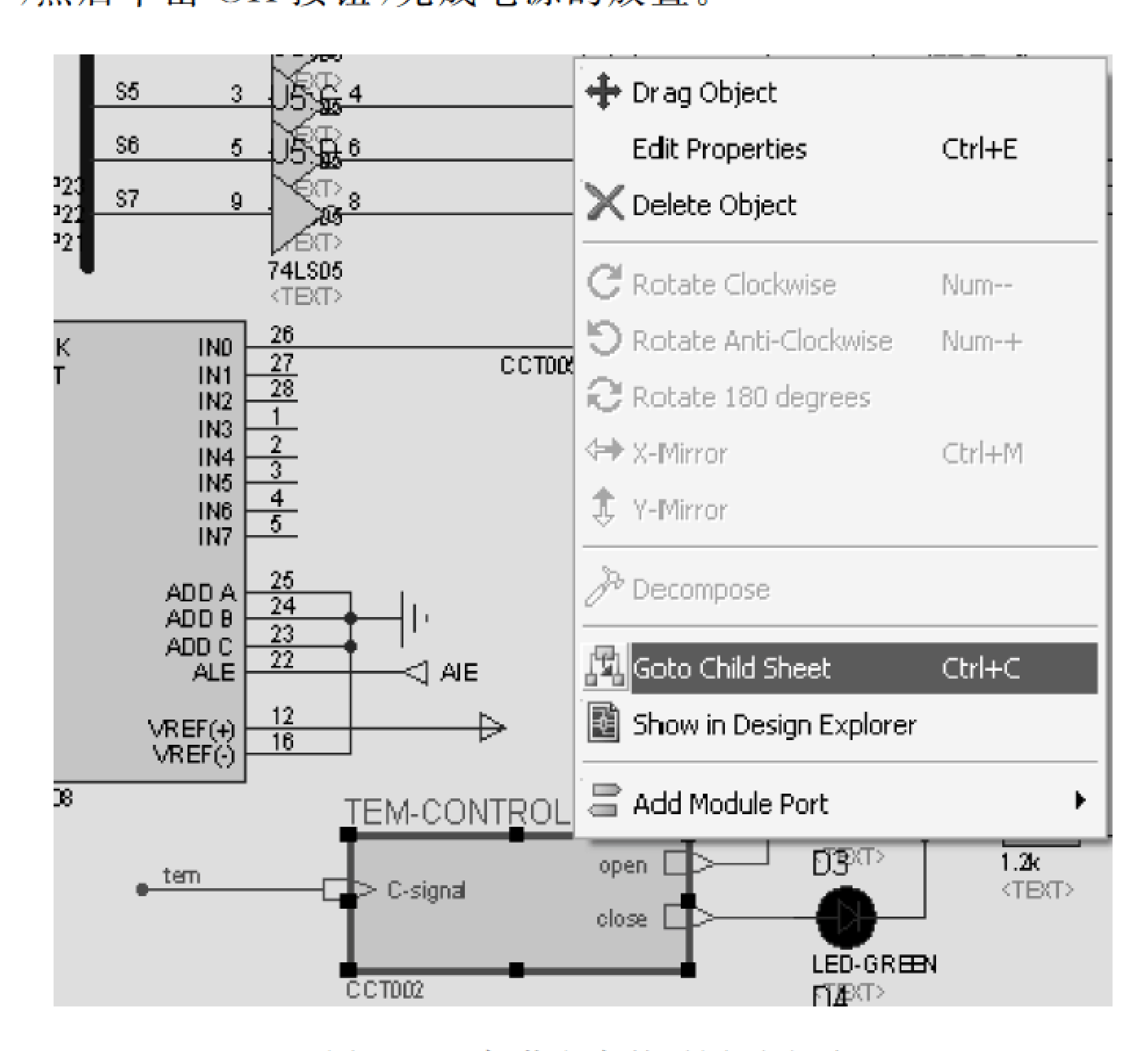


图 4.27　加载空白的子图页选项

输入/输出终端是必须进行放置的。选中对象编辑器中的INPUT/OUTPUT，则在预览窗口出现输入/输出端口的图标，在原理图中单击，则可在原理图中添加此端口；然后选中输入/输出端口符号，拖到合适的位置，并将其连接到电路；再次单击输入/输出端口符号，进入编辑对话框，在String栏中可分别输入端口名称，然后单击OK按钮，完成端口的放置，如图4.19中输入端口C-signal和输出端口open、close所示。

注意：这里的端口名称必须与子电路模块中一致。

4）返回主设计页

子电路编辑完后，选择菜单命令Design→Goto Sheet，这时出现如图4.28所示对话框，选择Root sheet 1，然后单击OK按钮，即使ISIS回到主设计图页。需要返回主设计页也可以在子图页空白处单击鼠标右键，选择Exit to Parent Sheet选项。

5）编辑子电路属性

双击子电路模块，进入子电路编辑对话框，可在Properties中输入子电路图属性。然后单击OK按钮，完成对该子电路的编辑，同时实现了电路的层次化。

6）子电路模块的复制

在层次电路图设计时，如果有多个子电路其内部电路相同，可通过子电路模块的复制得到，但需要注意的是，各子电路内部的各元器件标号必须是不同的，不能重复，所以，复制得到的子电路中元器件的标识需要重新进行排布。如图4.18中两个子电路模块TEM-CONTROL和FAN-CONTROL内部电路相同，但内部元器件的标号不能重复。可选择Tools→Global Annotator菜单项，打开全局标注器对话框如图4.29所示。其中，Scope为标注范围。系统提供了两种标注范围：Whole Design（整个设计）和Current Sheet（当前电路）；Mode为标注模式，系统提供了两种模式：Total（综合式）和Incremental（增量式）。这里可以选择Whole Design和Total，然后单击OK按钮，系统自动完成对复制得到的子电路的标注。



图4.28　Goto Sheet对话框

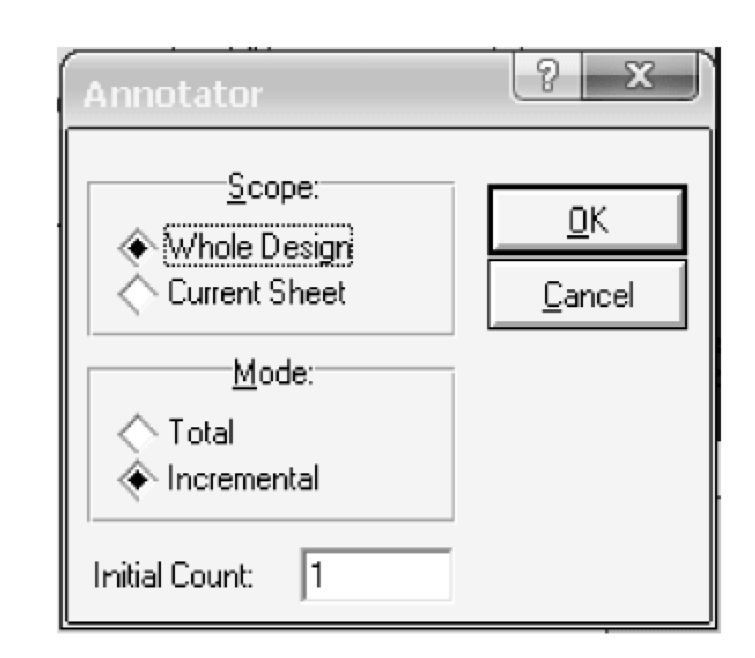


图4.29　全局标注器对话框

创建好子电路后，将其放到主电路中合适的位置，并将其与主电路相关部分相连，即可完成层次电路的设计。

4.4 系统仿真

本章的主要学习目的就是使设计人员掌握一个高效、便捷的软件工具，下面就以一个温度控制系统为例，介绍一下系统仿真的基本方法。

4.4.1 系统仿真举例

为便于构成一个完整的系统，控制器采用单片机来实现。要求设计一个基于单片机控制器的微型计算机温度测控系统；可以随时进行温度的设定，实时监测当前温度，并可进行实时控制；设定温度和当前温度要求用数码管显示；控制算法可采用模糊控制算法；采用Pt100温度传感器或DS18B20进行温度测量，测温范围为0～100℃；控制精度为±0.5℃。

根据以上设计要求，依据4.3节的原理图设计方法，完成如图4.30所示的一个基于单片机的温度测控系统原理图，用鼠标左键单击仿真运行按钮 ▶ ▮▶ ▮▮ ■ 中的 ▶ ，即可进入仿真运行状态。

从图4.30上可直观地看到数码管所显示设定温度为25℃，当前温度为35℃，降温指示灯(D4)点亮；另外图中tem和fan两个信号波形也可通过示波器进行观察，如图4.31所示为信号fan的波形。

在原理图设计的过程中，在调用电路元件时，为了能够实现仿真，要选用具有动画演示功能的器件或具有仿真属性的器件。如图4.32所示拾取元件时，在所选元件原理图符号窗口显示有“No Simulator Model”，则表示此元器件不具有仿真的功能；而像图4.33所示器件则具有有仿真属性，可以进行仿真。通过仿真，可实现声、光、动等逼真的效果，以检验电路硬件及软件设计的对错，非常直观，便于及时对电路进行修改。

另外，图4.30中的TEM-CONTROL、FAN-CONTROL和CELIANG都是采用层次电路设计的电路模块。TEM-CONTROL和FAN-CONTROL的内部电路如图4.19所示。CELIANG的内部电路如图4.34所示，其中包含一个热电阻Pt100，用于进行温度的测量。当然软件仿真的过程中，是不可能进行温度的测定的，但可以根据需要随意对当前温度进行设定，如图4.34中设定当前温度为35℃。

此外，仿真电路中还用到了一些仿真工具，如示波器、电压表、发光二极管等。下面对一些常用的仿真工具做一简单介绍。

4.4.2 常用仿真工具

Proteus有两种不同的仿真方式，即交互式仿真和基于图表的仿真。交互式仿真可以实时、直观地反映电路仿真运行的结果；基于图表的仿真用来精确分析电路的各种性能，如频率特性、噪声特性等。为辅助仿真运行，Proteus软件提供了大量的可视化虚拟设备，主要包括以下几项。

- 激励源：提供多种激励信号源。
- 虚拟仪器：用于观测电路的运行状况。
- 探针：直接布置在线路上，用于采集和测量电压或电流信号。
- 曲线图表：用于分析电路的波形、曲线、参数指标等。

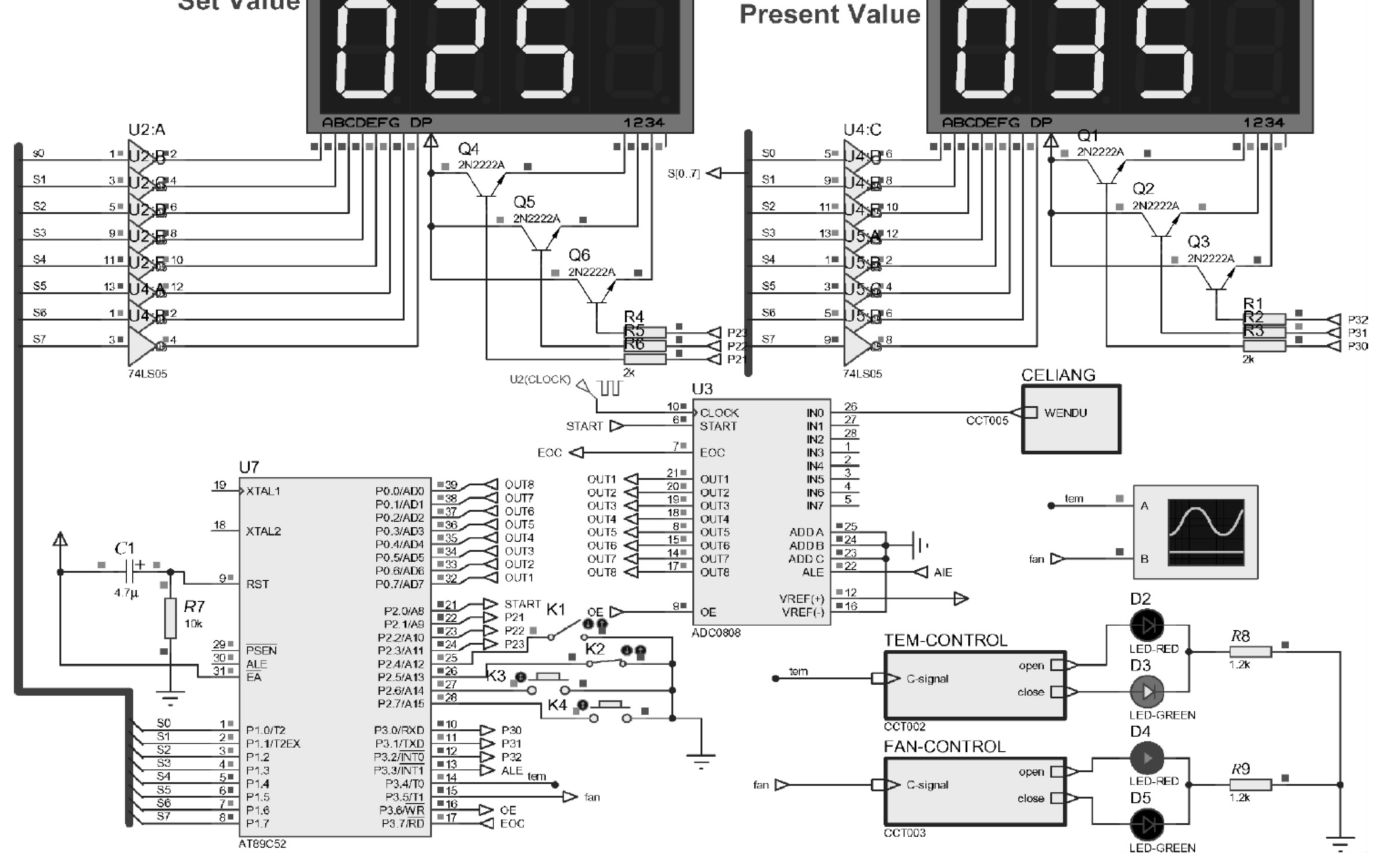


图 4.30　使用 Proteus 的 ISIS 所设计的温控系统原理图

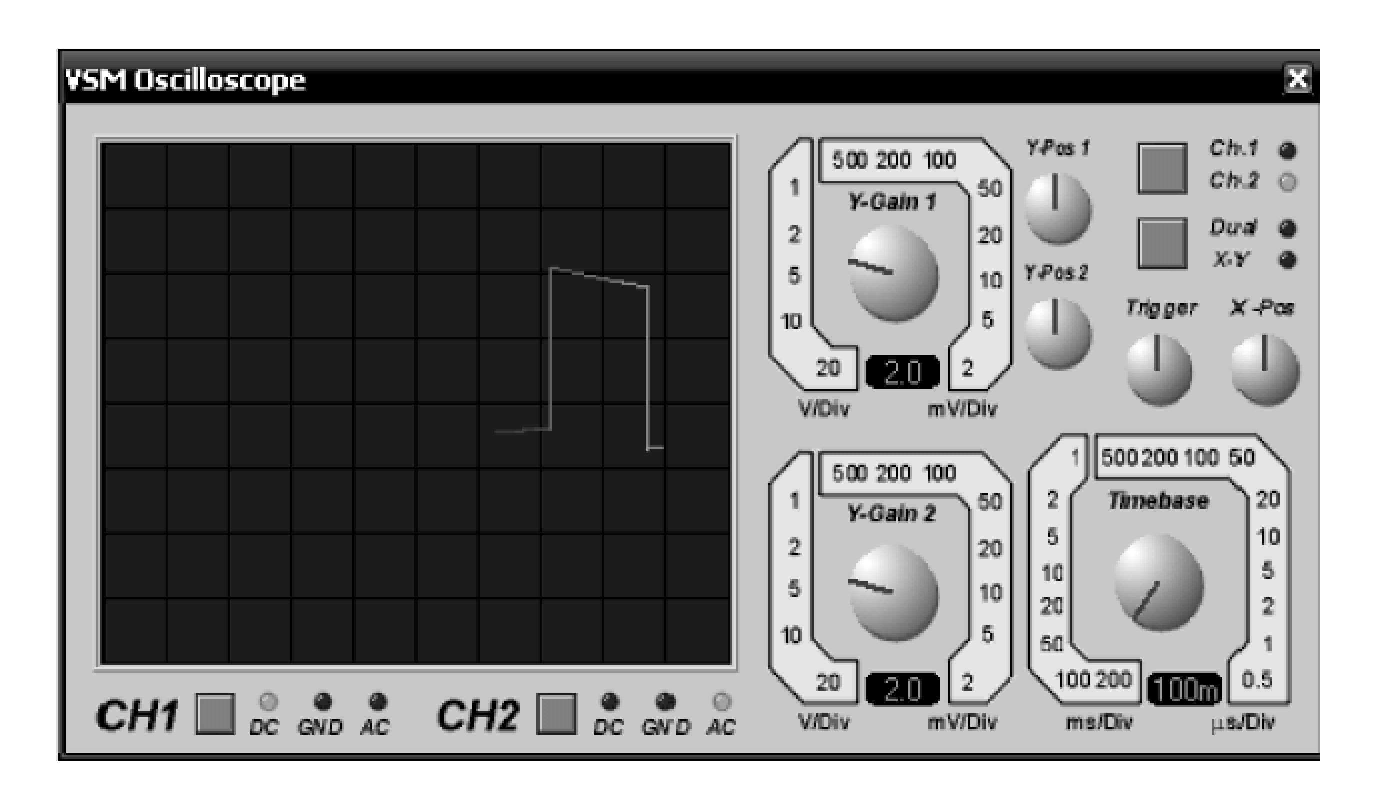


图 4.31 示波器显示

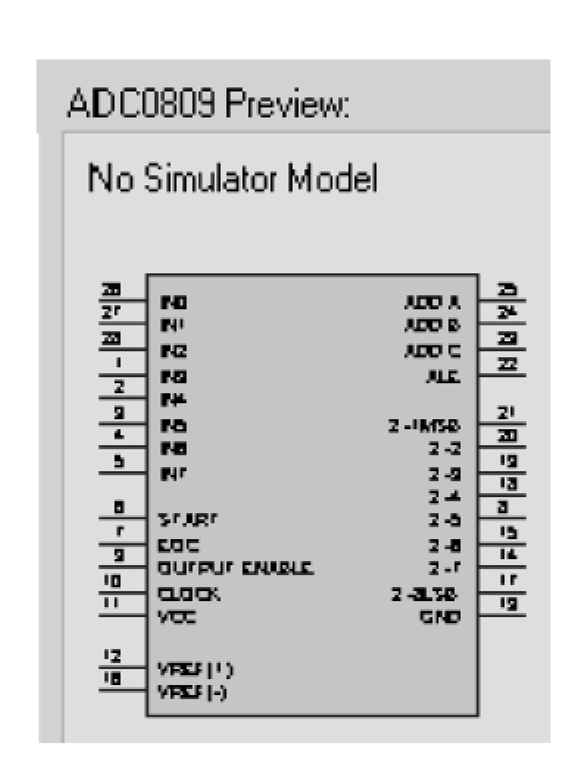


图 4.32 不具有仿真属性的元器件

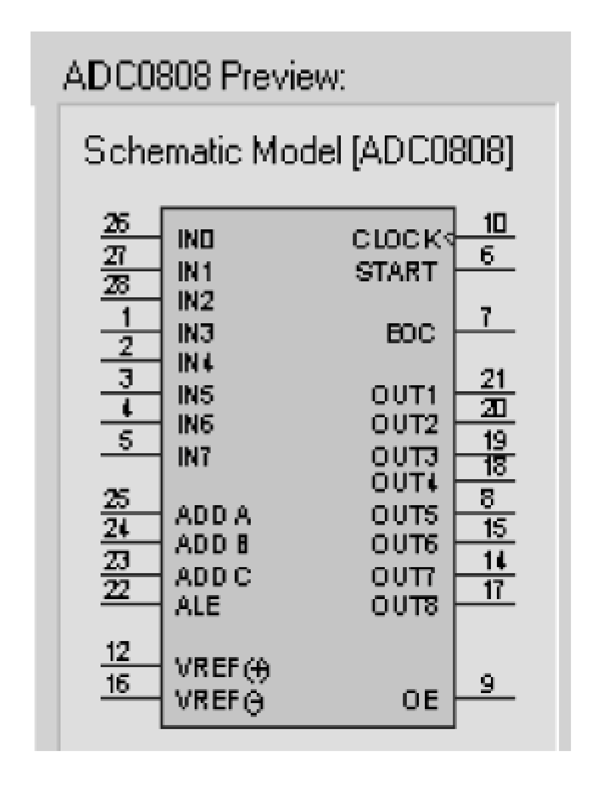


图 4.33 具有仿真属性的元器件

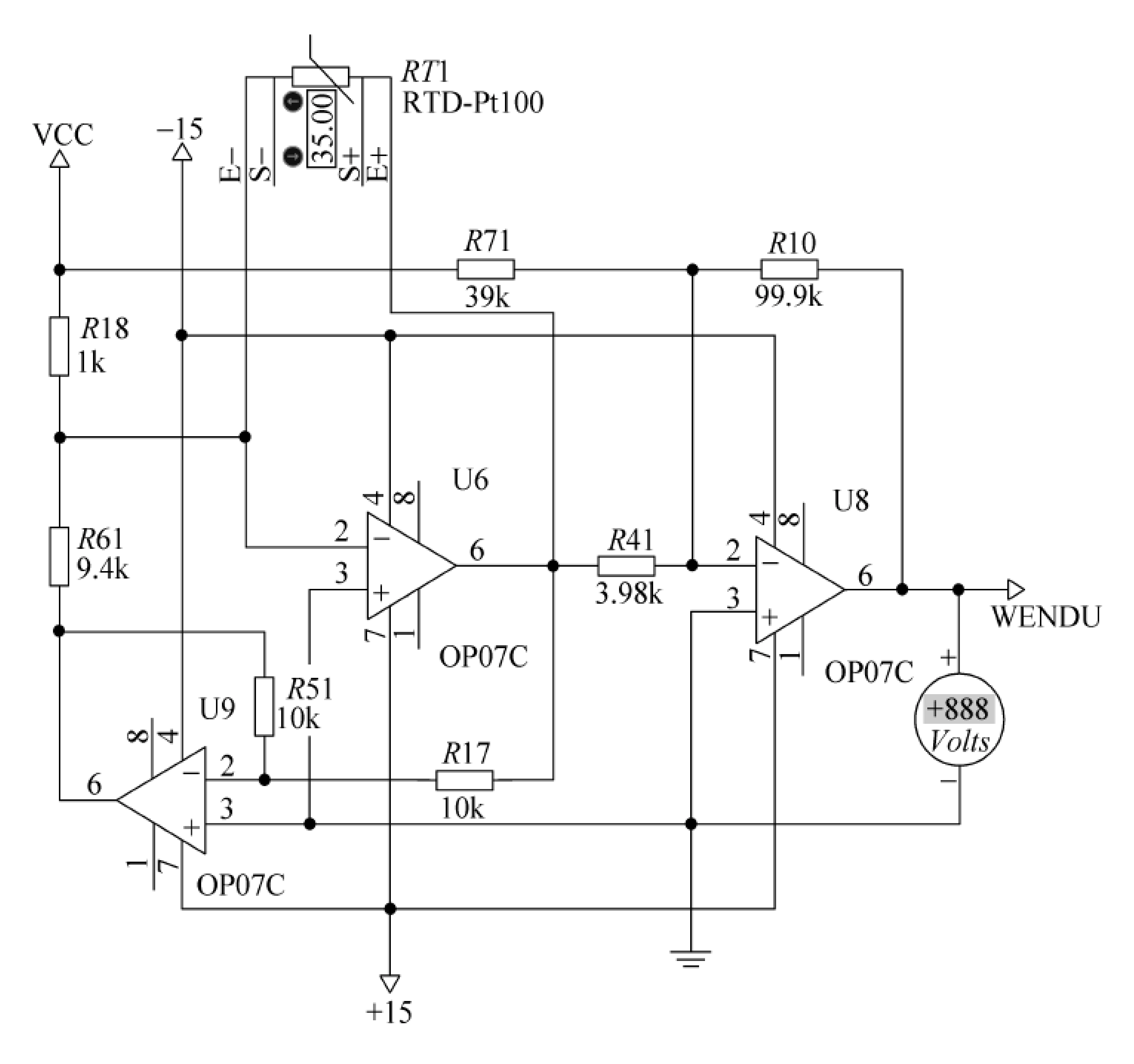


图 4.34 CELIANG 子电路的内部电路

1. 激励源

菜单通过单击 ISIS 主界面左侧的图标即可打开激励源，如图 4.35 所示。主要有以下一些激励源。

- DC：直流电压源。
- SINE：正弦波发生器。
- PULSE：脉冲发生器。
- EXP：指数脉冲发生器。
- SFFM：单频率调频波信号发生器。
- PWLIN：任意分段线性脉冲信号发生器。
- FILE：File 信号发生器，数据来源于 ASCII 文件。
- AUDIO：音频信号发生器，数据来源于 wav 文件。
- DSTATE：单稳态逻辑电平发生器。
- DEDGE：单边沿信号发生器。
- DPULSE：单周期数字脉冲发生器。
- DCLOCK：数字时钟信号发生器。
- DPATTERN：模式信号发生器。
- SCRIPTABLE：脚本信号发生器，采用 EasyHDL 语言的源代码描述信号发生器所需产生信号。

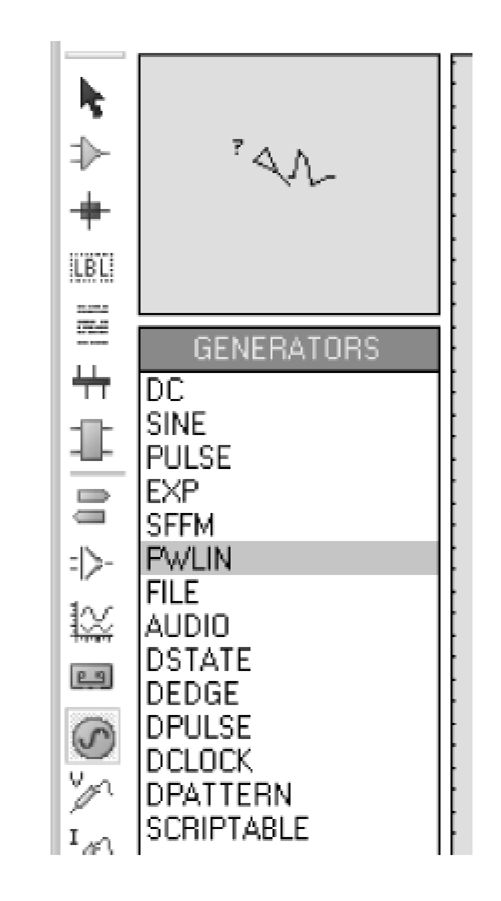
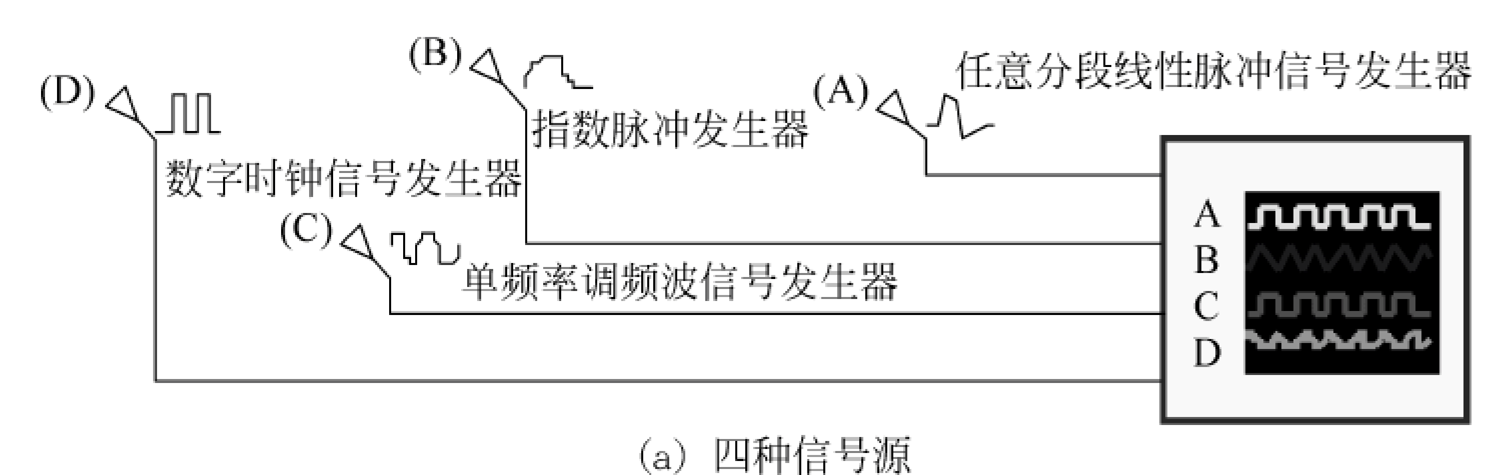
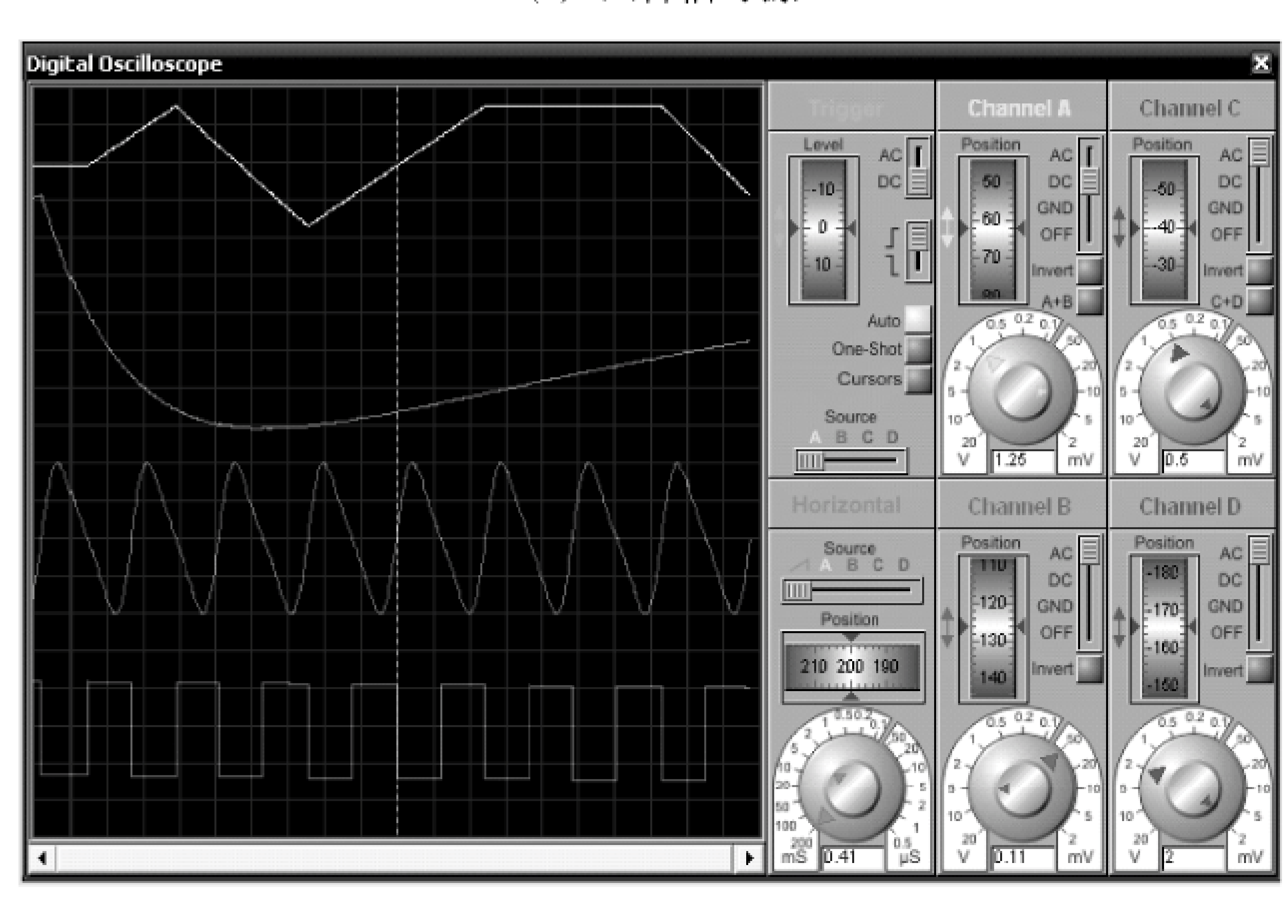


图 4.35　激励源列表

例如图 4.36(a)所示为四种信号源，包括数字时钟信号发生器、单频率调频波信号发生器、指数脉冲信号发生器和任意分段线性脉冲信号发生器；图 4.36(b)为用示波器观察到



(a) 四种信号源



(b) 用示波器观察到的信号

图 4.36　四种信号源及示波器观察到的相应信号

的波形。具体每一种信号源的属性都可以将鼠标放在信号图标上,单击鼠标右键,选择编辑属性(Edit Properties),然后进行编辑。

2. 虚拟仪器

虚拟仪器菜单通过单击 ISIS 主界面左侧的虚拟仪器模式(Virtual Instrument Mode)图标 即可打开,如图 4.37 所示。

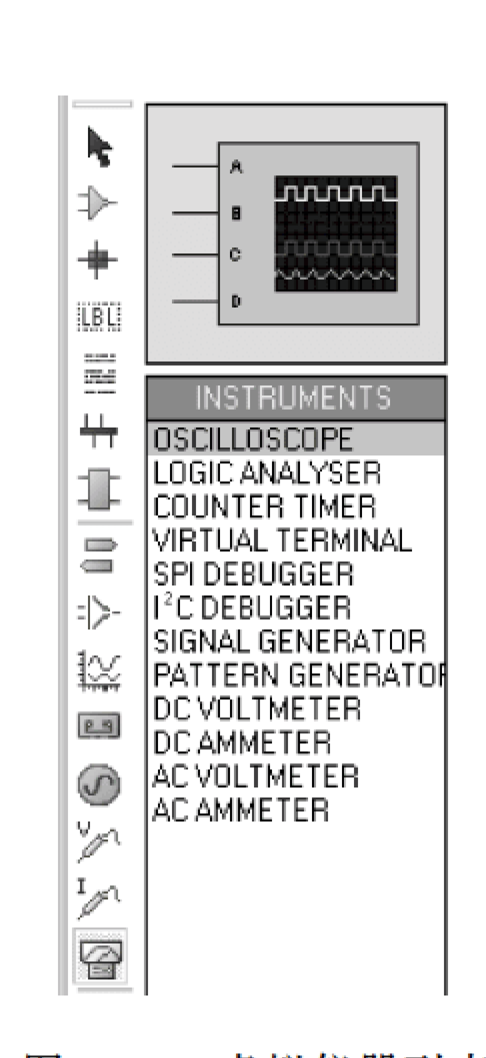


图 4.37 虚拟仪器列表

主要有以下一些虚拟仪器。

- OSCILLOSCOPE:示波器。
- LOGIC ANALYSER:逻辑分析仪。
- COUNTER TIMER:计数器/定时器。
- VIRTUAL TERMINAL:虚拟终端。
- SPI DEBUGGER:SPI 调试器。
- I^{2}C DEBUGGER:I^{2}C 调试器。
- SIGNAL GENERATOR:信号发生器。
- PATTERN GENERATOR:模式发生器。
- DC VOLTMETER:直流电压表。
- DC AMMETER:直流电流表。
- AC VOLTMETER:交流电压表。
- AC AMMETER:交流电流表。

下面介绍一下几种常用的虚拟仪表。

1) 示波器

虚拟示波器是最为常用的一种仪器,在 Proteus ISIS 环境中单击按钮图标 ,则可出现如图 4.37 所示的所有虚拟仪器名称列表。用鼠标左键单击列表区的 OSCILLOSCOPE,则在预览窗口出现示波器的符号,在编辑窗口单击鼠标左键,出现示波器的拖动图像,挪动鼠标到合适位置,再次单击左键,示波器被放置到原理图编辑区中去,如图 4.36(a)所示。

将所选虚拟示波器连入电路,当对电路仿真时,即可弹出如图 4.36(b)所示的示波器窗口,它的用法与实际示波器相仿,但可同时观察四路波形。示波器的四个接线端 A、B、C、D 应分别接四路输入信号,信号的另一端应接地。

2) 虚拟终端

虚拟终端(VIRTUAL TERMINAL)如图 4.38 所示,它相当于键盘和屏幕的双重功能,使用户在用到与上位机之间的串行通讯时,可直接由虚拟终端经 RS-232 模型进行数据的异步发送或接收。虚拟终端在运行仿真时会弹出一个仿真界面,如图 4.38 右下角所示,当由上位机向单片机发送数据时,可以和实际的键盘关联,用户可以从键盘经虚拟终端输入数据;当上位机接收到单片机发送来的数据后,虚拟终端相当于一个显示屏,会显示相应信息。

3) 计数器/定时器

计数器/定时器(COUNTER TIMER)的原理图符号及测试电路连线如图 4.39 所示。(CLK)为外加的时钟输入。

该仪器有如下三个输入端。

- CLK:计数和测频状态时,数字波的输入端。

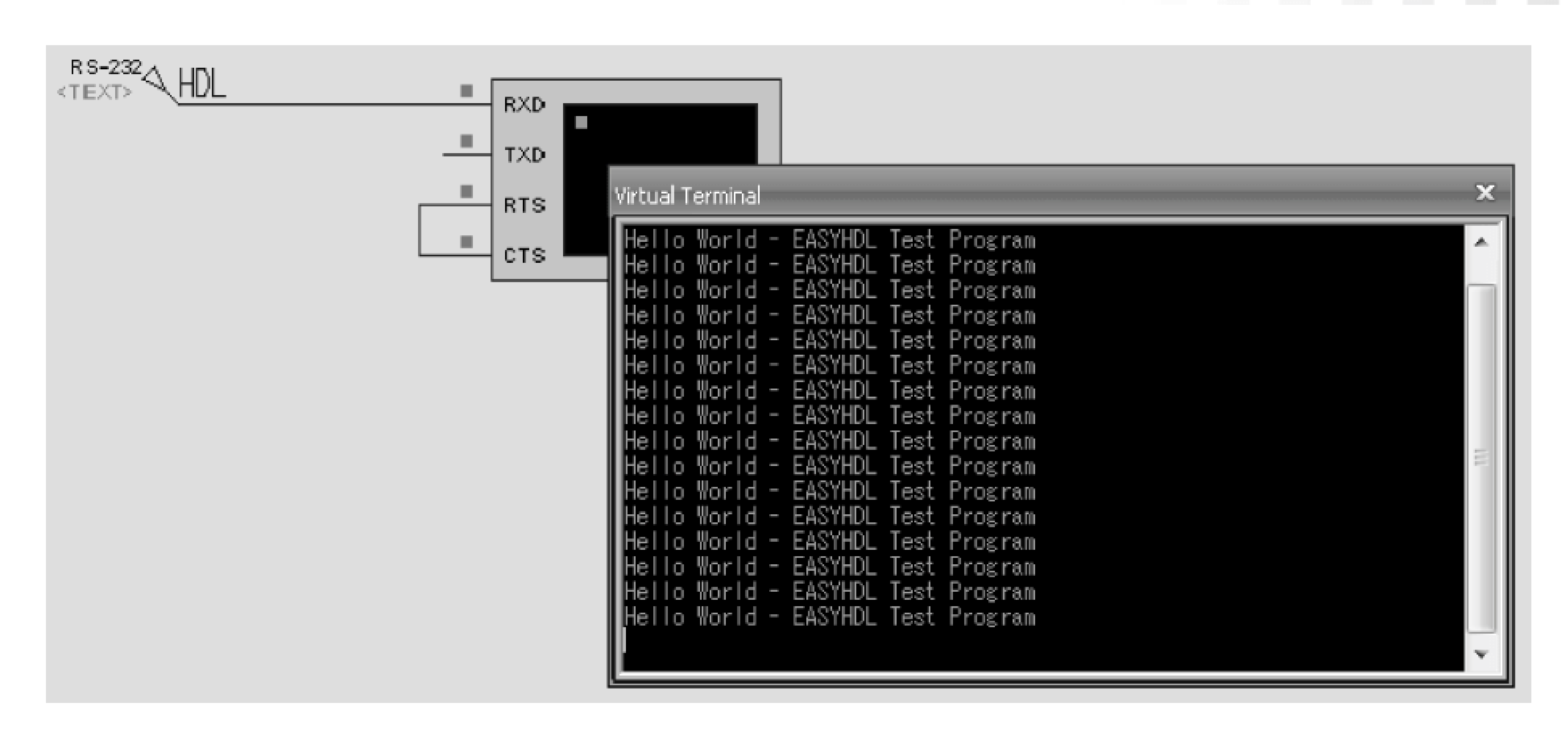


图 4.38　虚拟终端

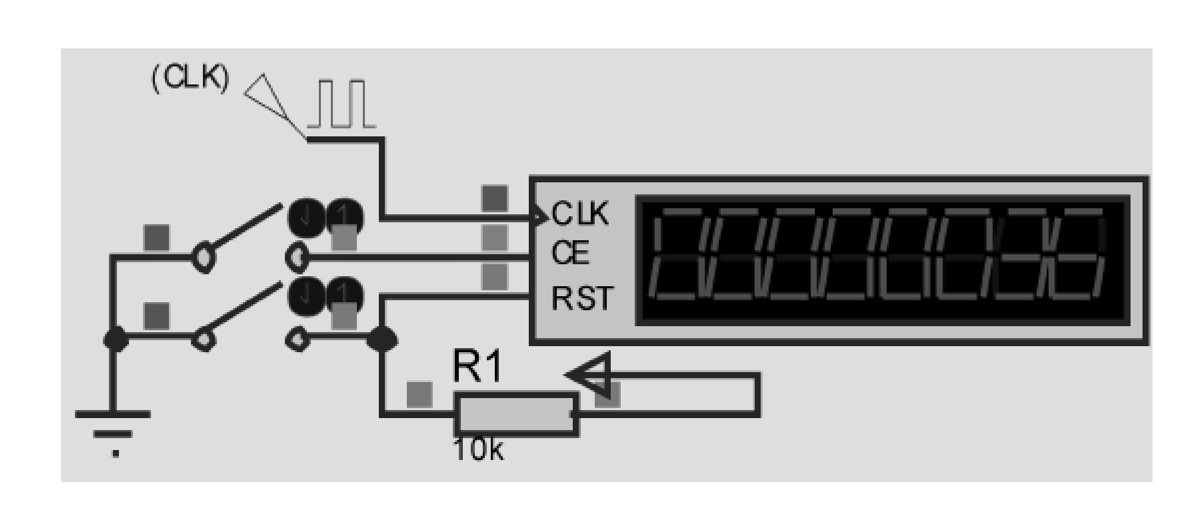


图 4.39　计数器/定时器电路

• CE：计数使能端(Counter Enable)，可通过计数器/定时器的属性设置对话框设为高电平或低电平有效，当此信号无效时，计数暂停，保持目前的计数值不变，一旦 CE 有效，计数继续进行。

• RST：复位端(RESET)，可设为上升沿有效(Low-High)或下降沿有效(High-Low)，当有效沿到来时，计时或计数复位到 0，然后立即从 0 开始计时或计数。

该仪器有四种工作方式，可通过属性设置对话框中的 Operating Mode 来选择，如图 4.40 所示。

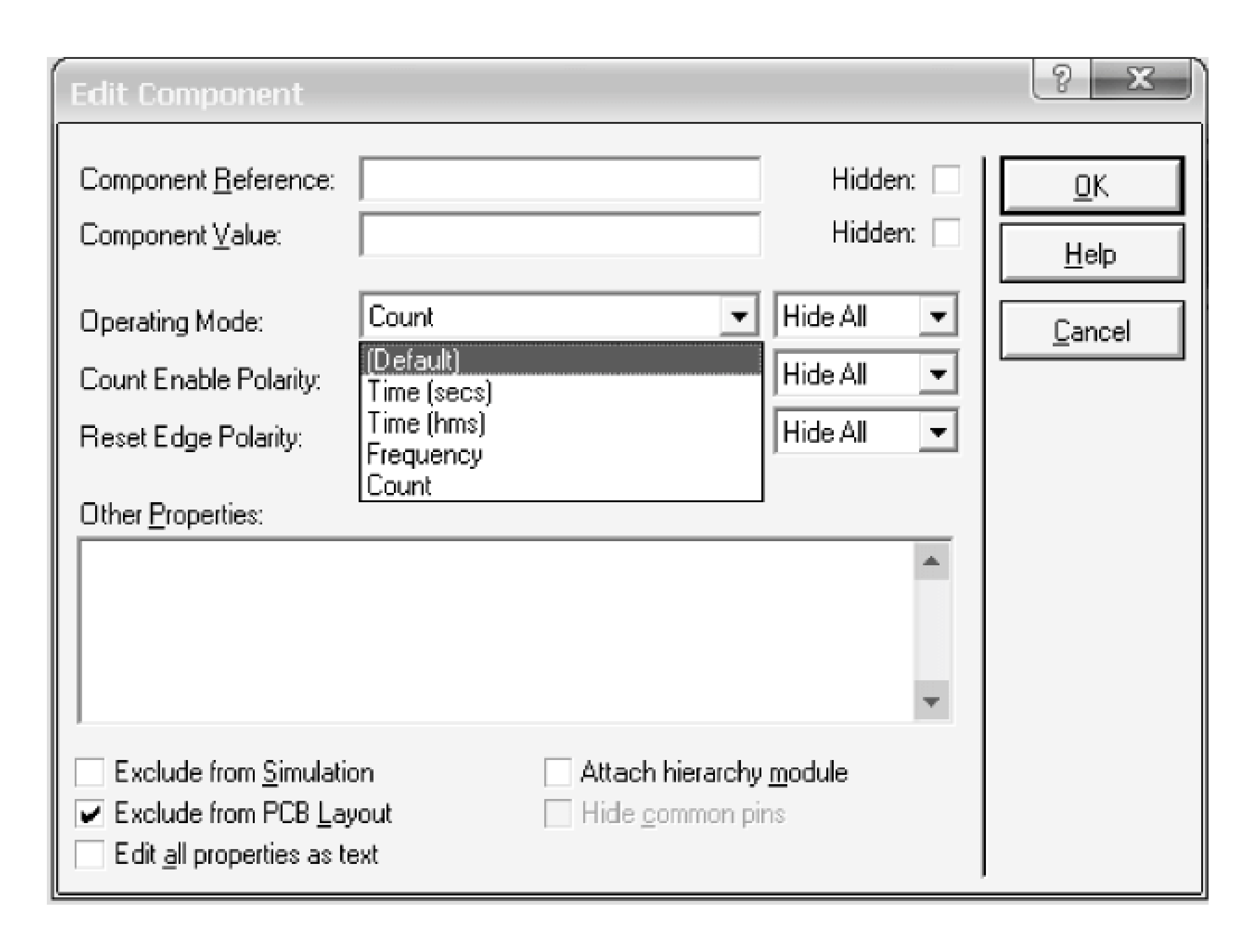


图 4.40　计数器/定时器的工作方式设置

- [Default]：缺省方式，系统设置为计数方式。
- Time[secs]：定时方式，相当于一个秒表，最多计 100s，精确到 1μs。CLK 端无需外加输入信号，内部自动计时。由 CE 和 RST 端来控制暂停或重新从零开始计时。
- Time[hms]：定时方式，相当于一个具有小时、分、秒的时钟，最多计 10h，精确到 1ms。CLK 端无需外加输入信号，内部自动计时。由 CE 和 RST 端来控制暂停或重新从零开始计时。
- Frequency：测频方式，在 CE 有效和 RST 没有复位的情况下，能稳定显示 CLK 端外加的数字波的频率。
- Count：计数方式，能够计外加时钟信号 CLK 的周期数，如图 4.39 中的计数显示，最多计满 8 位。

例如图 4.39，双击计数器/定时器，打开其属性设置对话框，如图 4.40 所示，设操作模式为 Count 即计数方式，计数使能端设为 High 高电平有效，复位端设为 Low-High 即上升沿有效，运行仿真，可显示如图 4.39 所示的计数方式，合上图中与 CE 相接的开关，则计数停止，打开开关则继续计数；合上与 RST 相接的开关再打开(即提供一个上升沿)，计数器清零后从零重新开始计数。

4) SPI 调试器

SPI 调试器的图形符号如图 4.41 所示。SPI 允许 MCU 与各种外围设备以同步串行通信方式交换信息。使用 SPI 调试器时，可将某一具体数值写入到 SPI 调试器中，同时也可以将 SPI 中存放的数据读出来。

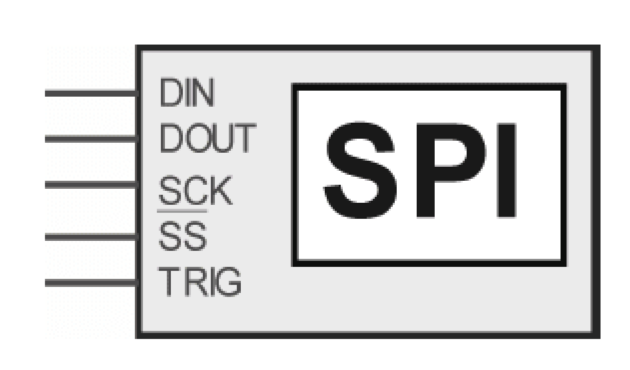


图 4.41 SPI 调试器

此元件共有五个接线端。

DIN：接收数据端；

DOUT：输出数据端；

SCK：连接总线时钟端；

$\overline{\text{SS}}$：从模式选择端，从模式时此端口为低电平才能使终端响应，主模式且当数据正传输时此端为低电平；

TRIG：输入端，能够把下一个存储序列放到 SPI 的输出序列中去。

双击 SPI 的原理图符号，可以打开它的属性设置对话框进行相关设置，如图 4.42 所示。对话框主要参数如下。

- SPI Mode：有三种工作模式可选择，Monitor 为监控模式，Master 为主模式，Slave 为从模式。
- Master clock frequency in Hz：主模式的时钟频率(Hz)。
- SCK Idle state is：SCK 空闲状态为高或者低，选择一个。
- Sampling edge：采样边，指定 DIN 引脚采样的边沿，选择当 SCK 从空闲到激活状态，或从激活到空闲状态时使用。
- Bit order：位顺序，指定一个传输数据的位顺序，可先传送最高位 MSB，也可先传送最低位 LSB。

例如，将 SCK 和 DIN 引脚连接到电路的相应端，打开 SPI 调试器属性设置对话框进行参数设置，设 SPI 为从模式，时钟频率与外时钟一致。运行仿真，弹出 SPI 的仿真调试窗口，如图 4.43 所示，且接收的数据将显示在窗口。



图 4.42　SPI 属性设置对话框

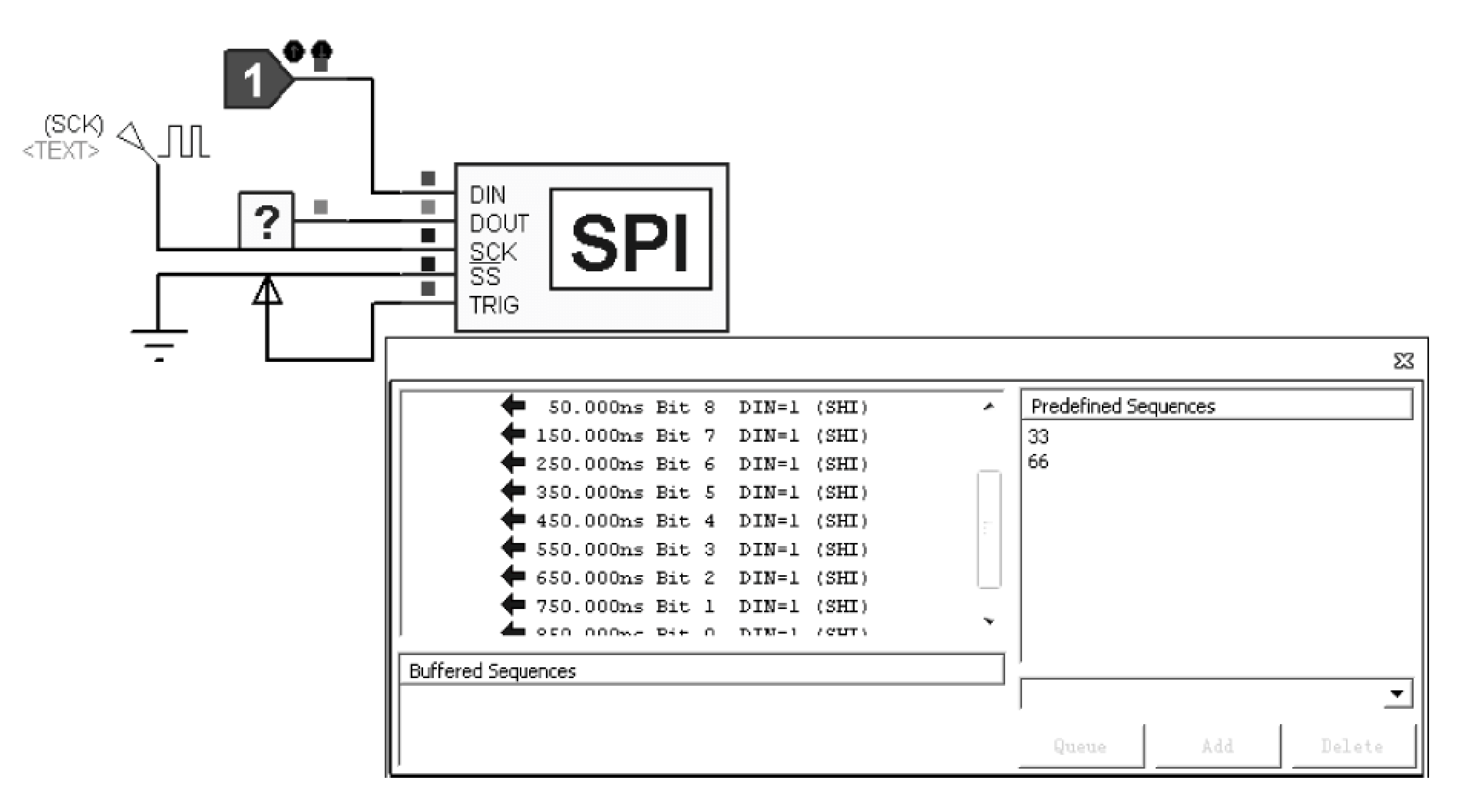


图 4.43　SPI 调试器的数据接收

如果把调试器设置为主模式，运行仿真，弹出 SPI 的仿真调试窗口。按下仿真按钮的暂停键 ▮▮ ，在调试窗口的右下方输入需要传输的数据，如“16”，按 Queue 按钮将输入的数据放入到数据传输队列 Buffered Sequences 中去，再次按仿真运行键，数据发送出去，数据发送完后，Buffered Sequences 清空，其上方的窗口显示发送信息，如图 4.44 所示。也可以按 Add 按钮把数据暂放到预传输序列中去备用，需要时加到数据传输队列中去。

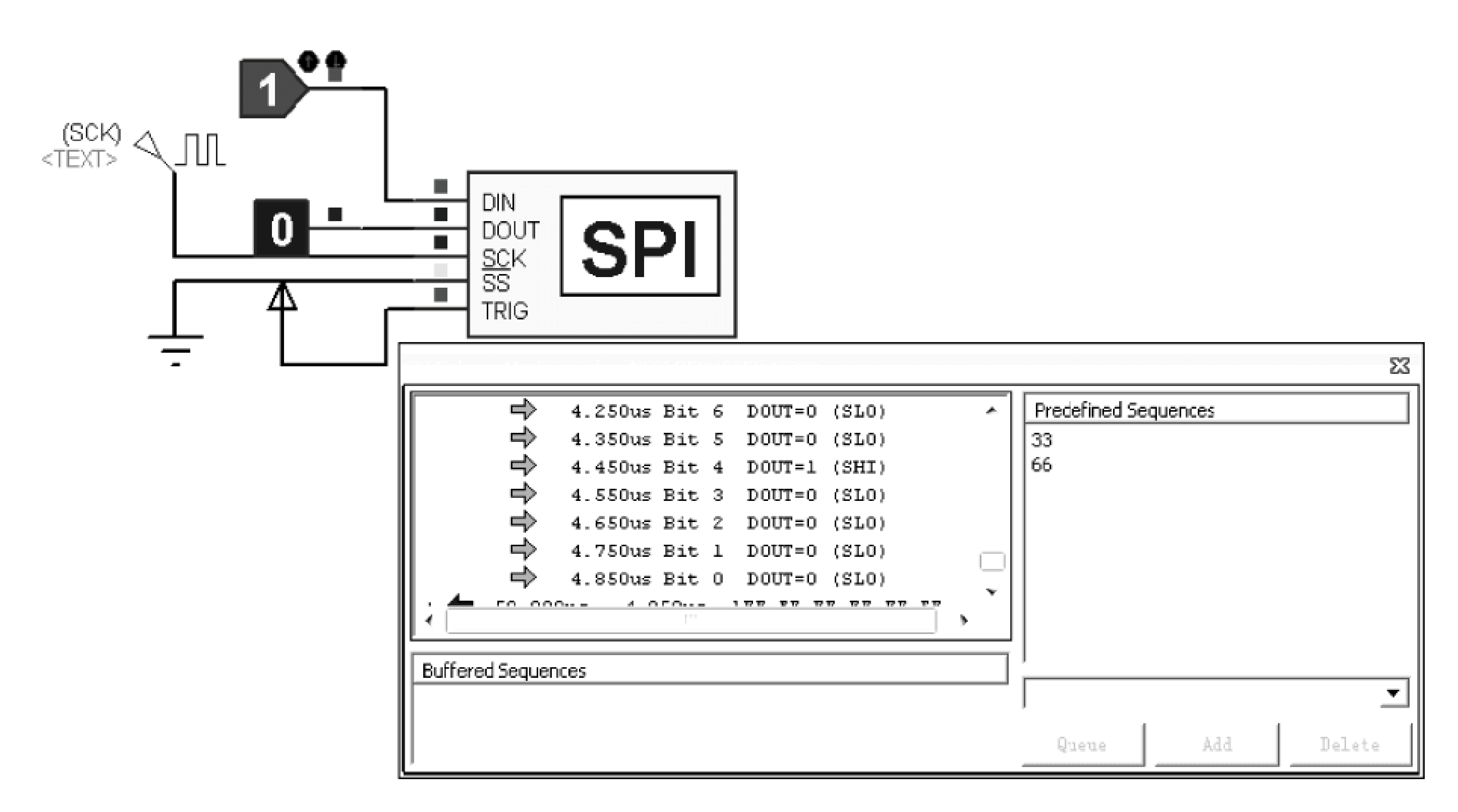


图 4.44 SPI 调试器的数据发送

5) I^{2}C 调试器

I^{2}C 调试器的图形符号如图 4.45 所示,它允许用户监测 I^{2}C 接口并与之交互,用户可以查看 I^{2}C 总线发送的数据,同时也可向总线发送数据。

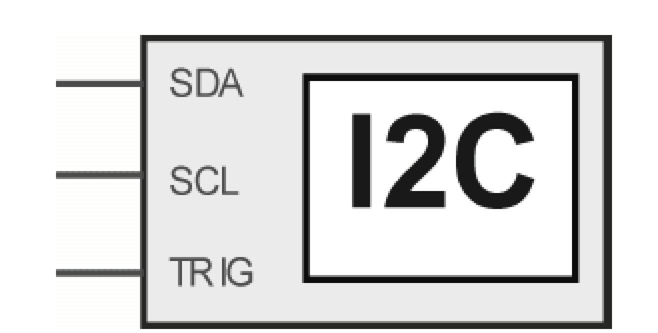


图 4.45 I^{2}C 调试器

I^{2}C 调试器共有三个接线端,SDA:双向数据线;SCL:双向端,连接时钟;TRIG:触发输入端,能使存储序列被连续地放置到输出队列中去。

双击该元件,打开属性设置对话框,主要参数如下。

- Address byte 1:地址字节 1,如果使用此终端仿真一个从元件时,则这一属性指定从器件的第一个地址字节。
- Address byte 2:地址字节 2,如果使用此终端仿真一个从元件时,并期望使用 10 位地址,则这一属性指定从器件的第二个地址字节。

I^{2}C 调试器的仿真运行界面与 SPI 类似。

除了上述虚拟仪器之外,还有直流电压表、直流电流表、交流电压表和交流电流表等,使用方法比较简单,这里不再详述。

3. 仿真图表

Proteus ISIS 除了可进行动态仿真以外,还提供一种静态的图表仿真功能,可通过放置电压/电流探针设置测试点,经仿真生成各个测试点的信号波形,即电路中某点对地的电压或某条支路的电流相对时间轴的波形,并以图表的形式留在电路图中,便于进行分析。图表仿真主要分为以下几个步骤。

(1) 在电路中各个被测点加电压探针,或在被测支路加电流探针。

(2) 选择放置波形的类别,并在原理图中拖出用于生成仿真波形的图表框。

(3) 在图表框中添加测量变量。

(4) 设置图表属性。

(5) 按图表仿真键生成对应的波形。

(6) 分析及打印输出。

1) 仿真图表应用举例

下面以图 4.46 所示 Proteus 软件自带的一个双极步进电机控制的仿真原理图为例,介绍仿真图表的应用。

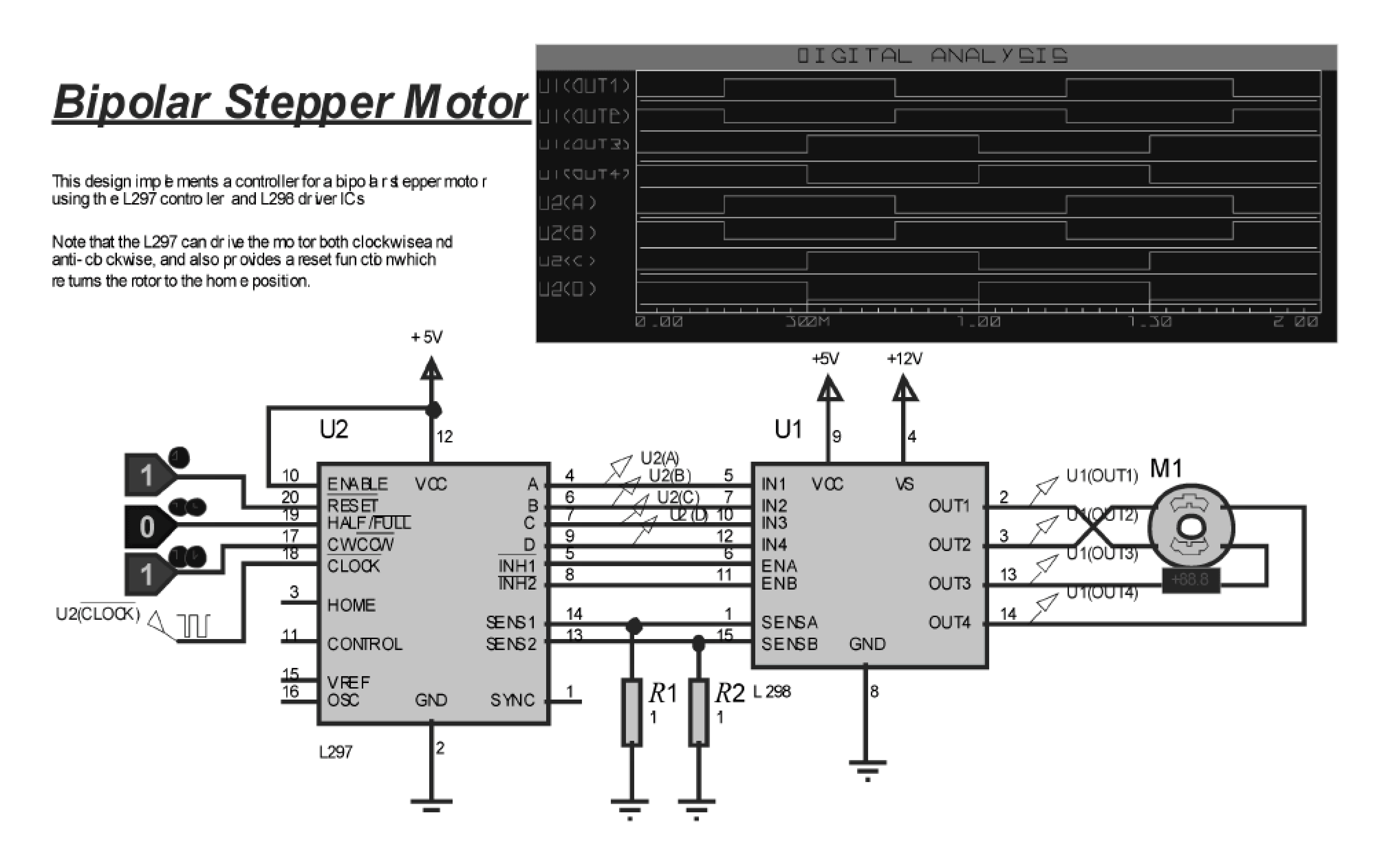


图 4.46　步进电机控制仿真原理图

图 4.46 中 L297 作为脉冲分配器,L298 为电机驱动器。其中,L297 的$\overline{\text{CLOCK}}$用于输入时钟信号,$\overline{\text{RESET}}$用于系统复位,HALF/$\overline{\text{FULL}}$进行电机的步距角选择,CW/$\overline{\text{CCW}}$进行电机转动方向的控制。为实现仿真,时钟信号取自时钟激励源,$\overline{\text{RESET}}$、HALF/$\overline{\text{FULL}}$和 CW/$\overline{\text{CCW}}$信号本应来自于控制器,这里因为主要是研究步进电机的运动情况,所以只用逻辑状态输入信号来代替(拾取元件时关键词为 LOGICSTATE),每次用鼠标左键单击它,可改变其逻辑状态。

按照以上说明建立好系统原理图之后,单击仿真运行键 ▶ ,即可看到步进电机的转动情况。改变 HALF/$\overline{\text{FULL}}$的逻辑状态,可看到步进电机每转动一步的步距角发生相应改变;改变 CW/$\overline{\text{CCW}}$的逻辑状态,电机的转动方向会随着翻转;修改时钟信号$\overline{\text{CLOCK}}$的频率,可改变电机的转动速度。即动态仿真可非常直观地看到步进电机的运动情况。但是通过 L298 的 OUT1～OUT4 输出给步进电机的信号 U1(OUT1)～U1(OUT4)与电机的运转情况究竟有什么关系,必须利用 U1(OUT1)～U1(OUT4)的波形作进一步分析,即进行图表仿真。图表仿真的具体步骤如下。

(1) 用鼠标左键单击绘图工具栏中的电压探针 (Voltage Probe Mode),然后将光标移动到需要观测的信号导线上,出现一个小白"×"号时单击鼠标左键,即可放置一个电压探针,连续操作,可放八个电压探针到电路的相应位置。如图 4.46 所示,每个探针都有一个名字,也可以为其重新命名。

(2) 单击绘图工具栏中的图表模式按钮 (Graph Mode),在对象选择窗口中选择DIGITAL,如图4.47所示。然后在图形编辑窗口中用鼠标拖出一个大小适中的图表框,如图4.48,系统默认横坐标是从0~1s。

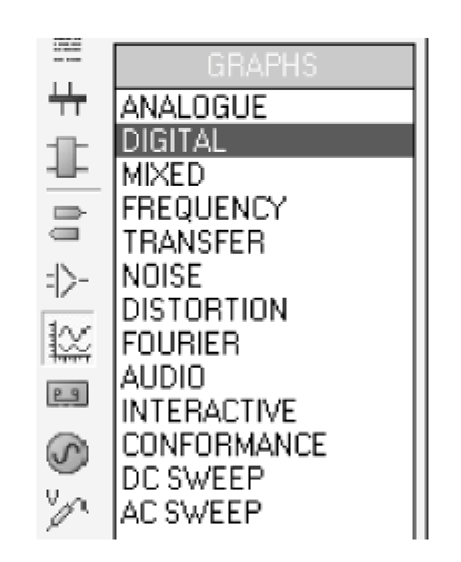


图4.47 图表仿真波形类别

(3) 将鼠标移至图表框,单击鼠标右键,在下拉窗口中选择Add Traces...,打开用于添加瞬态变量曲线窗口,如图4.49。在图4.49中,轨迹类型(Trace Type)下面的Digital被选中,表示需要添加的是数字变量。单击Probe P1的下拉箭头,出现如图4.50所示的所有探针名称。选中U1(OUT1),则该变量自动出现在Name栏中。重复同样步骤,即可添加所有要观测的变量,图4.51所示为添加完全部探针之后的仿真图表。

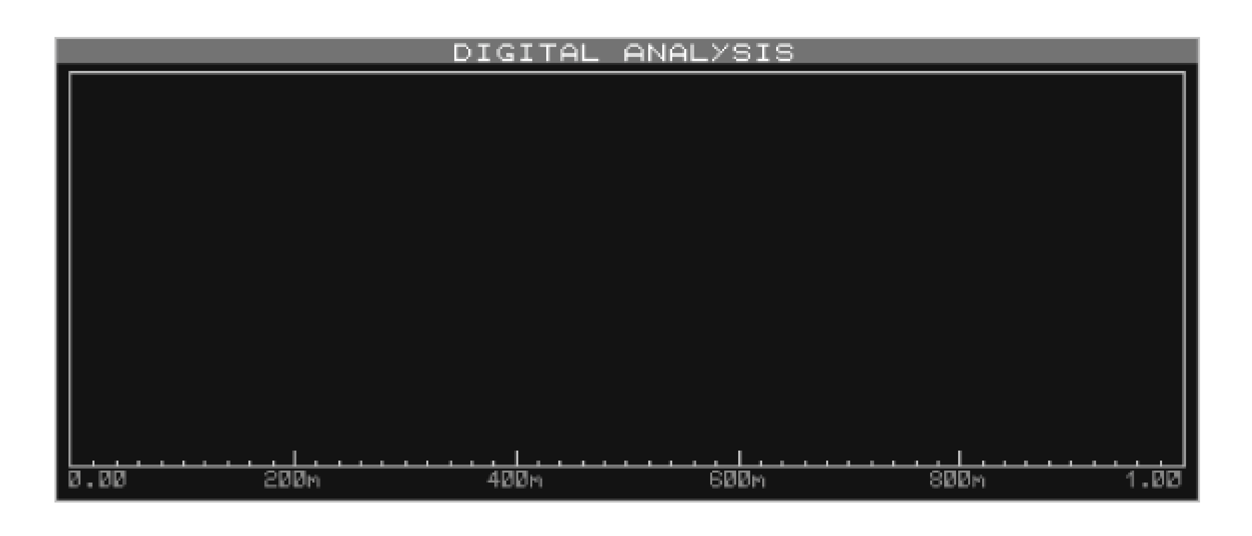


图4.48 数字波形仿真图表框

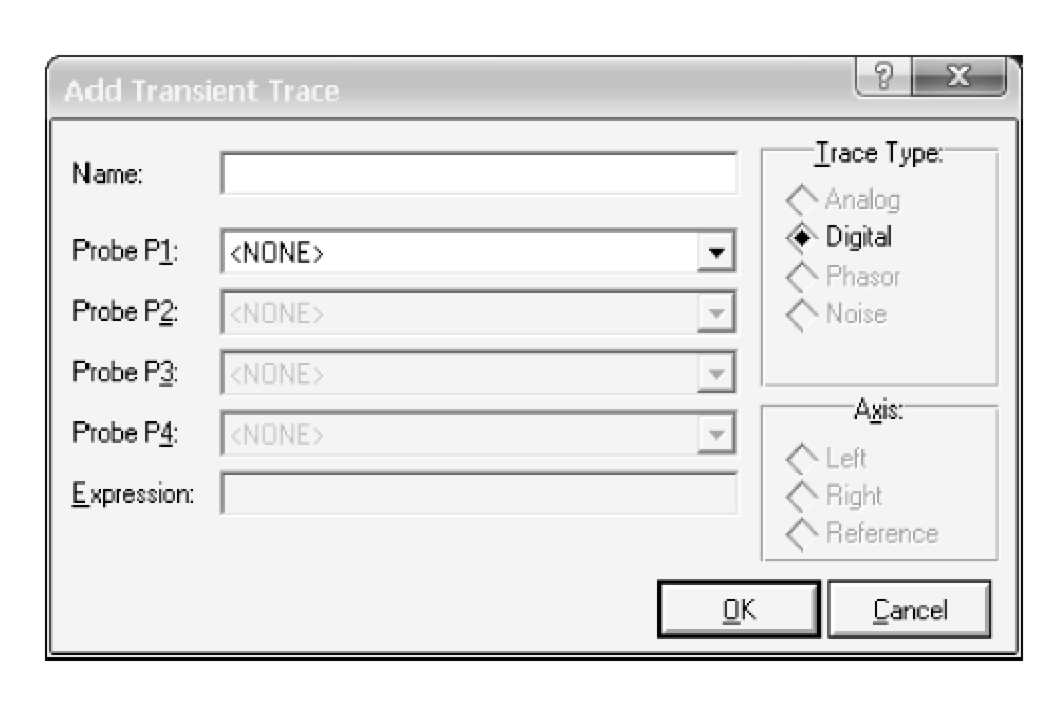


图4.49 添加瞬态变量曲线窗口

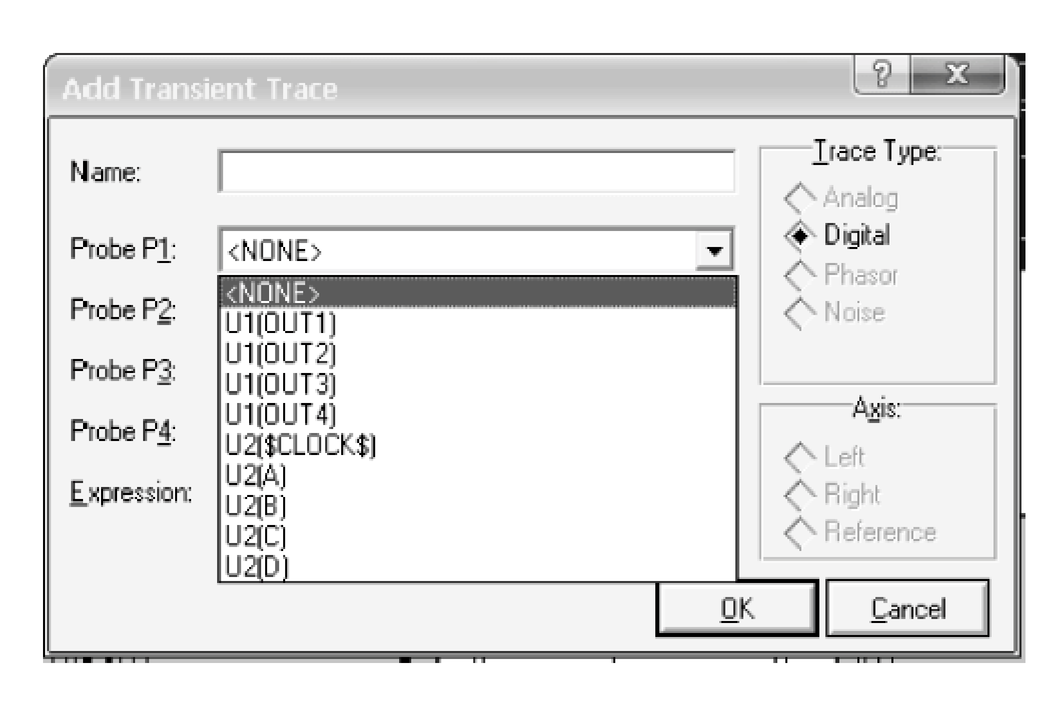


图4.50 添加瞬态变量

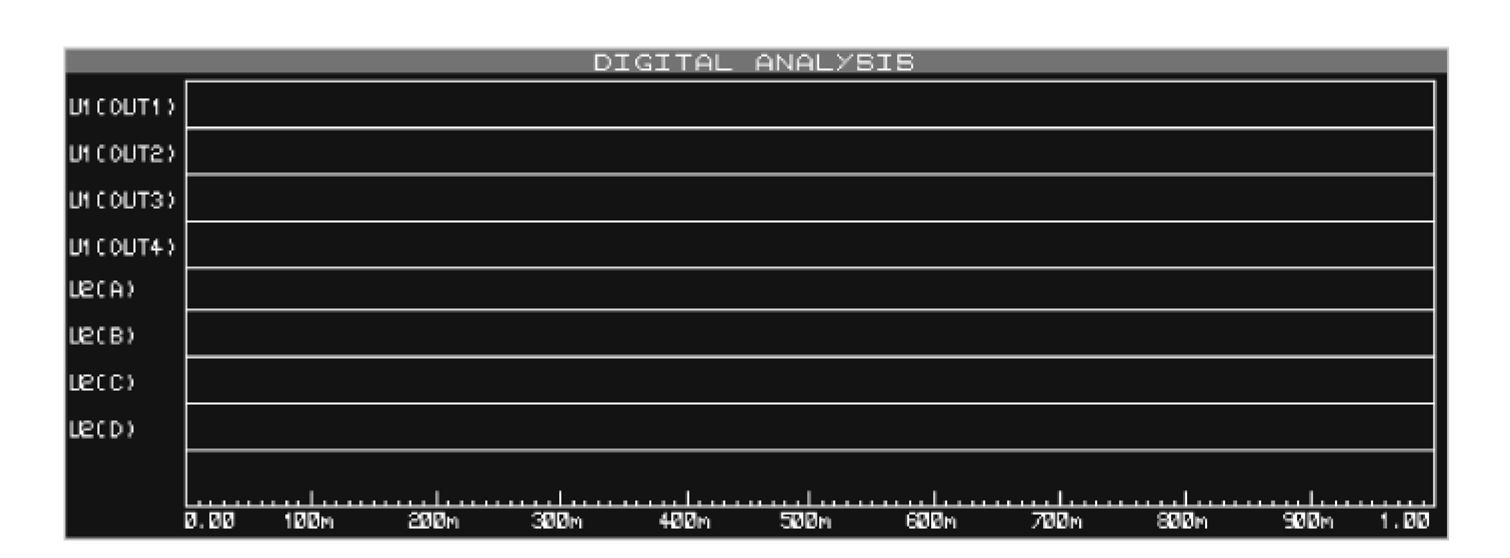


图4.51 添加完全部探针之后的仿真图表

(4) 双击图表,打开其属性对话框,可对其属性进行修改,如可将横坐标轴的时间范围做一修改,开始时间(Start time)不变,将结束时间(Stop time)改为2s,如图4.52所示。

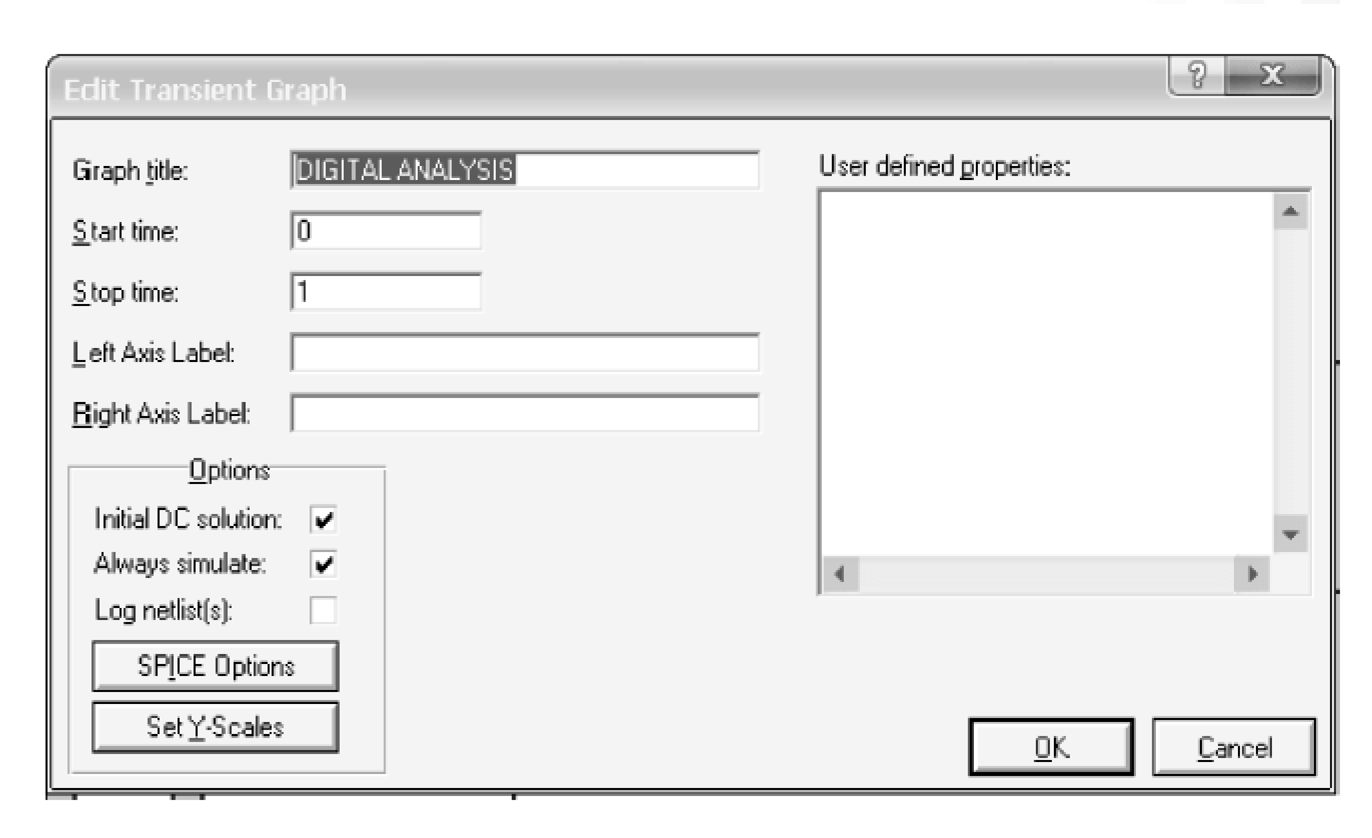


图 4.52　仿真图表属性对话框

(5) 再将鼠标移至图表框，单击鼠标右键，在下拉窗口中选择 Simulate Graph，即可得到方针后的图表，如图 4.53 所示。

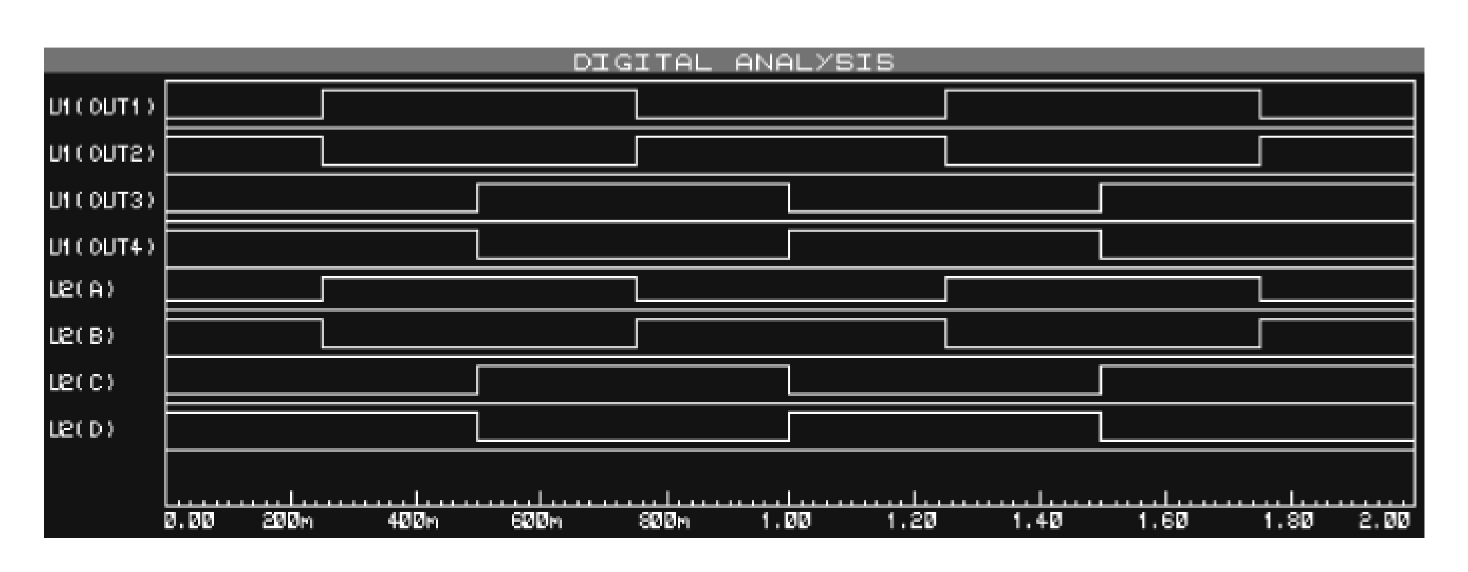


图 4.53　仿真后的图表

(6) 把鼠标放在图 4.53 图表框的绿色区双击，图表放大，成为一个独立的窗口，放大后的图表窗口最上面的菜单栏可用于对图表做进一步的设定，如可修改背景及图形颜色等，如图 4.54 所示。

通过对图 4.54 中曲线的分析，可清楚了解步进电机的输入信号与电机运动情况之间的关系。关于步进电机控制的相关知识将在第 7 章数字程序控制中详细介绍。

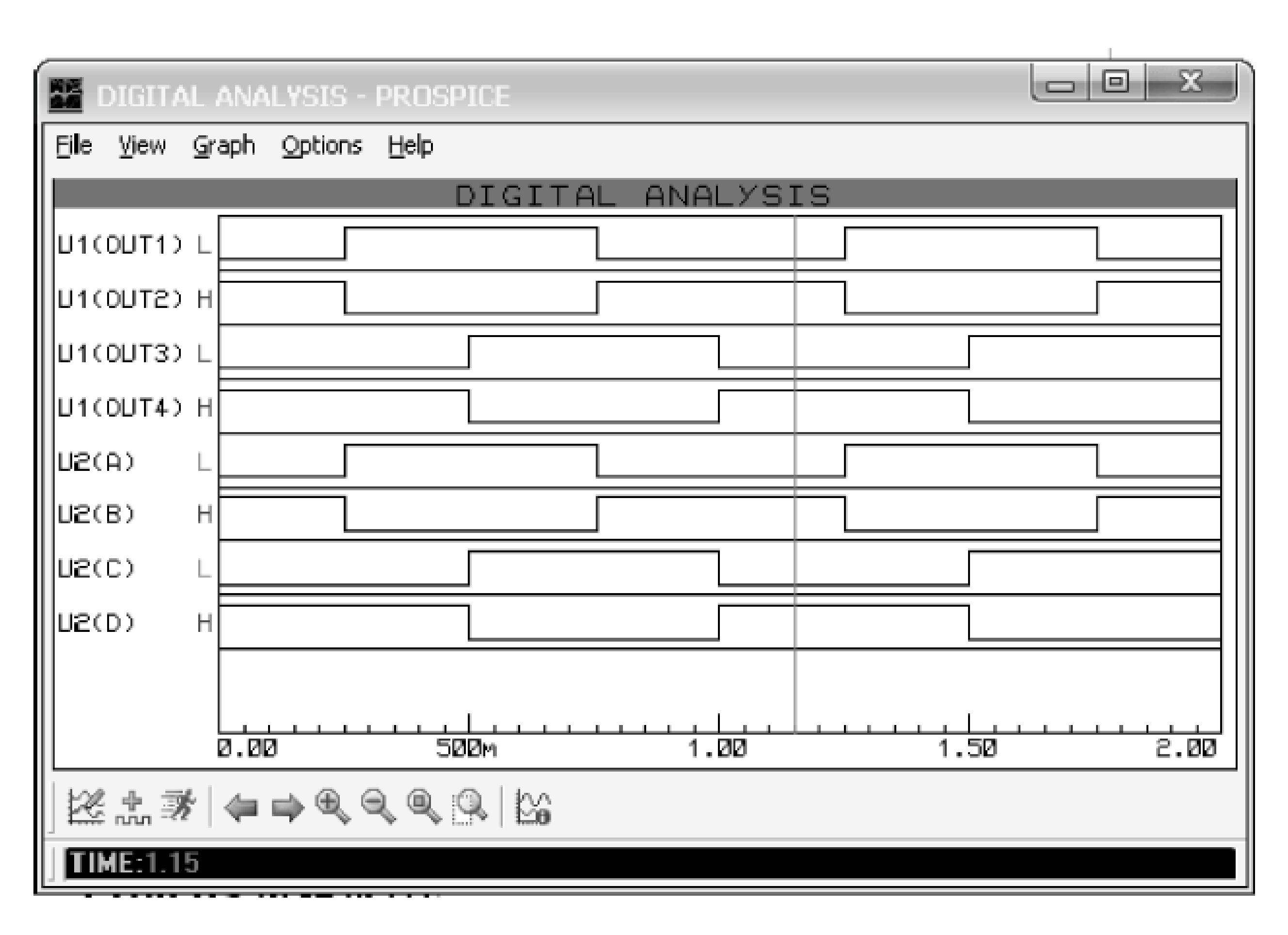


图 4.54　放大后的图表窗口

2）其他仿真图表介绍

鼠标左键单击绘图工具栏中的图表模式按钮(Graph Mode)时可以在对象选择窗口中看到如图 4.47 所示共有 13 类图表波形类别，具体见表 4.5 所列。

表 4.5 图表仿真波形种类

图表仿真波形名称	含　义	图表仿真波形名称	含　义
ANALOGUE	模拟波形	FOURIER	傅里叶分析
DIGITAL	数字波形	AUDIO	音频分析
MIXED	模数混合波形	INTERACTIVE	交互分析
FREQUENCY	频率响应	CONFORMANCE	一致性分析
TRANSFER	转移特性分析	DC SWEEP	直流扫描
NOISE	噪声分析	AC SWEEP	交流扫描
DISTORTION	失真分析		

需要绘制模拟波形时使用 ANALOGUE；需绘制数字波形时使用 DIGITAL；如果需要把两类波形放在同一个图表框中，则需要选定混合波形 MIXED。

习题

1. 针对第 3 章习题第 4 题：试用 AEN(DMA 地址使用信号，高电平有效)信号、地址信号 A_0～A_9、74HCT138 芯片，以及其他门电路，搭建一个地址译码电路，可寻址的地址分别为 3A0H～3A7H。搭建 Proteus 仿真电路，并进行仿真分析。

2. 建立仿真电路，对图 4.55 电路进行仿真分析。

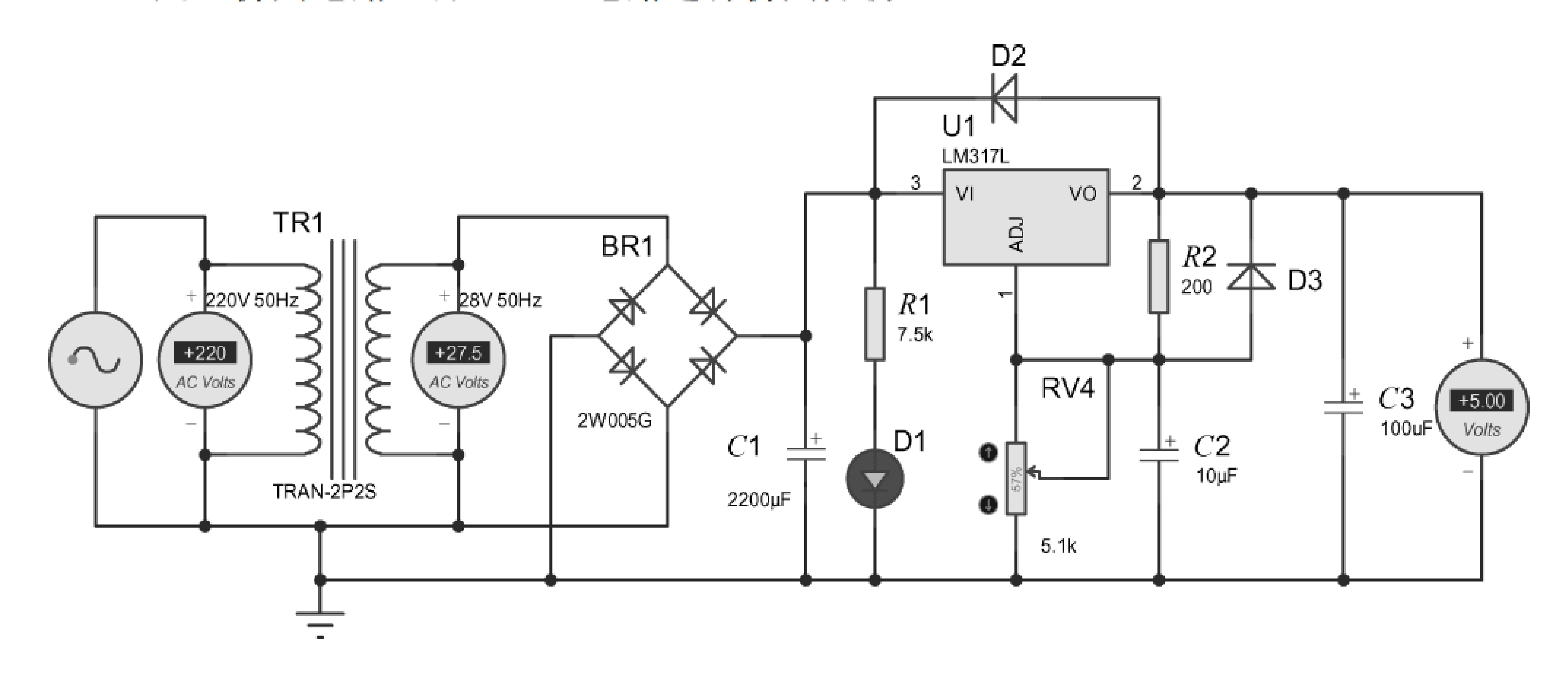


图 4.55　题 2 电路

3. 练习示波器的使用。

4. 搭建图 4.56 电路，并进行分析、仿真。

5. 自行设计一个电路，练习图表仿真方法。

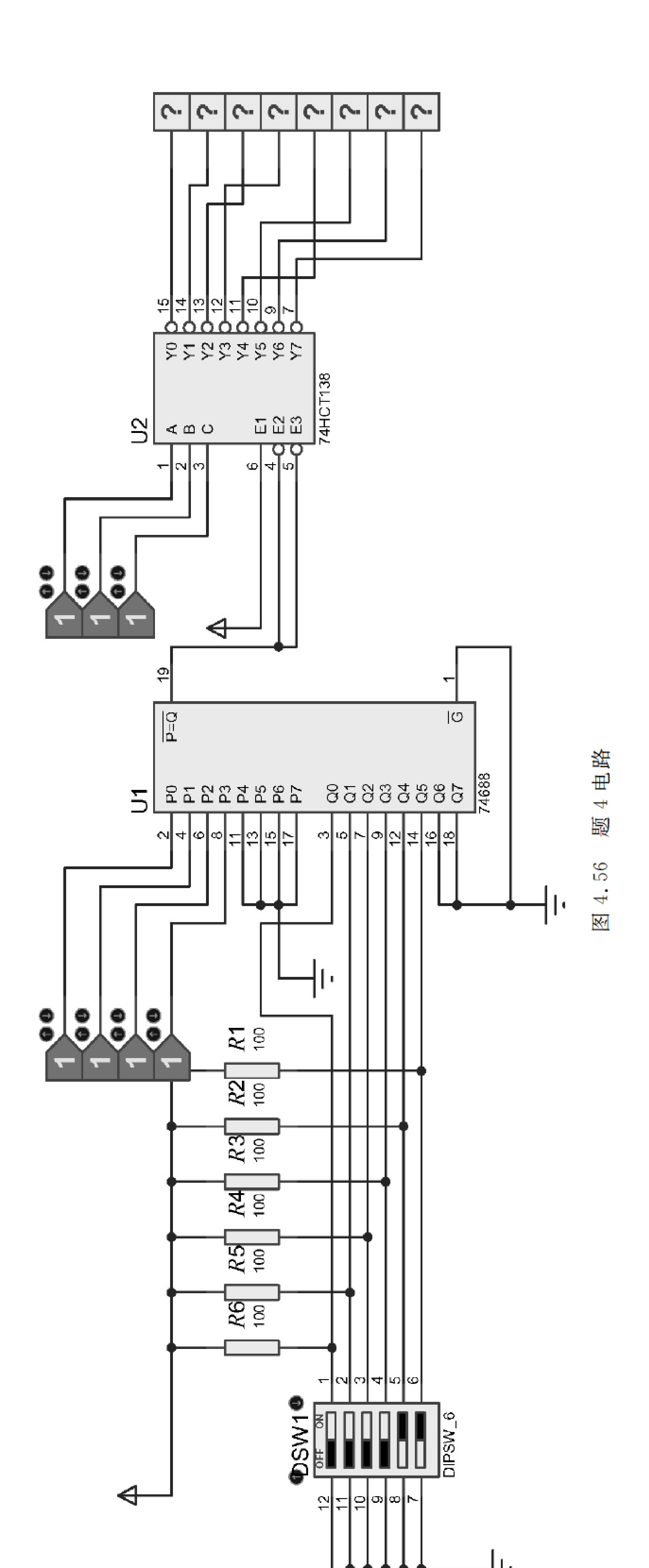

U1
74688
P0
P1
P2
P3
P4
P5
P6
P7
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
P=Q
G
U2
74HCT138
A
B
C
E1
E2
E3
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
R1 100
R2 100
R3 100
R4 100
R5 100
R6 100
DSW1
DIPSW_6
ON
OFF

图 4.56 题 4 电路

第5章 计算机控制系统的控制算法

计算机控制系统的设计，实际上是指数字控制器的设计，而并不是涉及计算机控制系统工程的全部设计内容。控制器的任务主要是按照一定的控制规律和相应的控制算法，来产生满足系统性能指标的控制信号，从而输出到现场以驱动执行机构，达到控制的目的。

在传统的模拟控制系统中，控制器的控制规律是由控制仪表或者电子电路等硬件设备来完成的，而在计算机控制系统中，除了硬件装置的计算机外，其设计核心主要体现在数字控制器的设计算法和实现上。

数字控制器的设计分为连续化设计（或称为模拟化设计）和离散化设计（或称为数字化设计）两种。本章首先介绍了分析离散系统的数学工具 Z 变换和 Z 反变换，着重阐述了数字控制器的连续化设计和离散化设计的基本原理和步骤，并详细介绍了最少拍控制系统的设计，同时对目前广泛使用的 PID 调节器的离散化方法、PID 数字控制器的算法改进以及参数整定进行了深入的讨论，最后简单介绍了史密斯预估控制和串级控制等复杂控制的规律。

5.1 线性离散系统的 Z 变换及 Z 反变换

连续系统输入量和输出量之间的关系可用微分方程来描述，借助于拉普拉斯变换建立系统的传递函数，可以非常方便地分析系统的特性。而计算机控制系统是一种采样控制系统，即离散系统，对于离散系统主要用差分方程来描述，并借助于所谓的 Z 变换来建立系统的脉冲传递函数，也可以方便地分析离散系统的性能。由于连续控制系统和离散控制系统的控制方法不同，所以它们使用的数学工具、研究方法也不同，表 5.1 列出了它们的研究方法对照表。

表 5.1 控制系统的研究方法对照表

研究方法分类	连续控制系统	离散控制系统
输入量与输出量之间的关系表达	微分方程	差分方程
数学工具	拉普拉斯变换	Z 变换
使用的函数	传递函数	脉冲传递函数
现代控制理论	状态方程	离散状态方程

Z 变换的最初思想来源于连续系统，它是由拉普拉斯变换直接引申出来的一种变换，实际上它是拉普拉斯变换的一种变形，有时又称为采样拉普拉斯变换，是研究离散系统的重要数学工具。

5.1.1 Z 变换

1. Z 变换的定义

将连续信号变为离散信号的过程称为采样过程，设连续函数 $f(t)$是符合拉普拉斯变换条件的，则采样后的信号 $f^*(t)$也可以进行拉普拉斯变换。连续信号 $f(t)$通过采样周期为 T 的理想采样后得到的采样信号 $f^*(t)$是一组加权理想脉冲序列，每个采样时刻的脉冲强度等于该采样时刻的连续函数值，即采样后其离散函数 $f^*(t)$为

$$f^*(t) = \sum_{k=0}^{+\infty} f(kT)\delta(t-kT) = f(t)\sum_{k=0}^{+\infty}\delta(t-kT) \tag{5.1}$$

式(5.1)中，T 为采样周期，k 为采样序号。

研究采样信号 $f^*(t)$可像连续函数一样对离散信号 $f^*(t)$采用拉普拉斯变换，可得

$$L[f^*(t)] = F^*(s) = L[f(t)\sum_{k=0}^{+\infty}\delta(t-kT)] = \sum_{k=0}^{+\infty} f(kT)e^{-kTs} \tag{5.2}$$

式(5.2)中右边为无穷多项之和，每一项中都有因子 e^{-kTs}，同时它是一个 s 的指数函数，因为复变量 s 在指数里不方便计算，无法使问题简化，因此引入一个新的复变量 z 为

$$z = e^{Ts} \tag{5.3}$$

或

$$s = \frac{1}{T}\ln z \tag{5.4}$$

代入式(5.2)中，可得

$$F^*(s)\,|_{s=\frac{1}{T}\ln z} = F(z) = \sum_{k=0}^{+\infty} f(kT)(e^{Ts})^{-k} = \sum_{k=0}^{+\infty} f(kT)z^{-k} \tag{5.5}$$

式(5.5)称做离散信号 $f^*(t)$的 Z 变换，并记为

$$F(z) = Z[f^*(t)] = \sum_{k=0}^{+\infty} f(kT)z^{-k} \tag{5.6}$$

所以，由上可见，所谓的 Z 变换，也就是在离散信号 $f^*(t)$的拉普拉斯变换中令 $z=e^{Ts}$ 而得到的变换，可以视为拉普拉斯变换的一种变形。式(5.6)是 $f^*(t)$的单边 Z 变换，若式中的流动变量 k 从 $-\infty \sim +\infty$，则称为双边 Z 变换。但是由于控制系统中研究的信号都是从研究时刻 $t=0$ 开始算起，所以使用的都是单边 Z 变换，在这里也简称为 Z 变换。

2. Z 变换的求法及简单函数的 Z 变换

Z 变换的求解方法就是求取采样函数的 Z 变换式，其求法有很多，比如级数求和法、部分分式法、留数计算法等，这里重点介绍级数求和法和部分分式法这两种比较常用的 Z 变换方法。

1) 级数求和法

级数求和法是直接应用 Z 变换的定义进行求和。设时间连续函数为 $f(t)$，其对应的离散时间函数为 $f^*(t)$，则其 Z 变换根据式(5.6)可表示为

$$F(z) = Z[f^*(t)] = \sum_{k=0}^{+\infty} f(kT)z^{-k}$$

将上式右边展开，即得

$$F(z)=\sum_{k=0}^{+\infty} f(kT)z^{-k}=f(0)+f(T)z^{-1}+f(2T)z^{-2}+\cdots+f(kT)z^{-k}+\cdots \tag{5.7}$$

这就是离散时间函数 $f^*(t)$进行 Z 变换的一种级数表达形式。由这种表达式可知，如果知道了时间连续函数为 $f(t)$在各个采样时刻 $kT(k=0,1,2,\cdots,\infty)$上的采样值 $f(kT)$，便可根据上式求得其 Z 变换的级数展开形式，它是无穷多项的级数，是开式，可以根据具体问题将这种开式简化为闭式以便于运算。

【例 5.1】 求指数函数 $f(t)=\mathrm{e}^{-aT}$ 的 Z 变换。

解： 首先求出函数 $f(t)$在各个采样时刻的采样值，即 $f^*(t)=f(kT)=\mathrm{e}^{-akT}$ 所以可根据式(5.7)写出

$$F(z)=1+\mathrm{e}^{-aT}z^{-1}+\mathrm{e}^{-2aT}z^{-2}+\cdots+\mathrm{e}^{-akT}z^{-k}+\cdots$$

上式两边同时乘以 $\mathrm{e}^{-aT}z^{-1}$，则得

$$\mathrm{e}^{-aT}z^{-1}F(z)=\mathrm{e}^{-aT}z^{-1}+\mathrm{e}^{-2aT}z^{-2}+\cdots+\mathrm{e}^{-(k+1)aT}z^{-(k+1)}+\cdots$$

再将两式相减，可得

$$F(z)(1-\mathrm{e}^{-aT}z^{-1})=1$$

整理可求得

$$F(z)=\frac{1}{1-\mathrm{e}^{-aT}z^{-1}}=\frac{z}{z-\mathrm{e}^{-aT}}$$

【例 5.2】 求单位斜坡函数 $f(t)=t$ 的 Z 变换。

解： 首先求出函数 $f(t)$在各个采样时刻的采样值，即 $f^*(t)=f(kT)=kT$，所以可根据式(5.7)写出

$$F(z)=\sum_{k=0}^{\infty} kTz^{-k}$$

由于

$$\sum_{k=0}^{\infty} z^{-k}=\frac{z}{z-1}$$

将上式两边对 Z 求导数，并将和式与导数交换得

$$\sum_{k=0}^{\infty}(-k)z^{-k-1}=\frac{-1}{(z-1)^2}$$

再将上式两边同时乘以$(-Tz)$，于是便可求得单位斜坡函数 $f(t)=t$ 的 Z 变换为

$$F(z)=\sum_{k=0}^{\infty} kTz^{-k}=\frac{Tz}{(z-1)^2}$$

2) 部分分式法

设时间连续函数为 $f(t)$，其对应的拉普拉斯变换为 $F(s)$。已知 $F(s)$，而 $f(t)$没有给出，可以直接通过 $F(s)$求出 $f(t)$的 Z 变换 $F(z)$。

部分分式法也即先将连续函数的拉普拉斯变换展开成部分分式 $F(s)=\sum_{i=1}^{n}\frac{A_i}{s+p_i}$ 之和的形式，然后对各个分式分别求其 Z 变换，其和即为所求的 Z 变换。

【例 5.3】 求解 $F(s)=\frac{a}{s(s+a)}$的 Z 变换。

解： 因为

$$F(s)=\frac{A}{s}+\frac{B}{s+a}=\frac{1}{s}-\frac{1}{s+a}$$

所以原函数为

$$f(t)=L^{-1}F(s)=1(t)-\mathrm{e}^{-aT}$$

因此可得其 Z 变换为

$$F(z)=Z[1(t)]-Z[\mathrm{e}^{-aT}]=\frac{z}{z-1}-\frac{z}{z-\mathrm{e}^{-aT}}=\frac{z(1-\mathrm{e}^{-aT})}{(z-1)(z-\mathrm{e}^{-aT})}$$

【例 5.4】 求解 $f(t)=\sin\omega T$ 的 Z 变换。

解：因为

$$L[f(t)]=L[\sin\omega T]=\frac{\omega}{s^2+\omega^2}=\frac{-\frac{s}{2\mathrm{j}}+\frac{\omega}{2}+\frac{\omega}{2}+\frac{s}{2\mathrm{j}}}{s^2+\omega^2}=\frac{-\frac{1}{2\mathrm{j}}}{s+\mathrm{j}\omega}+\frac{\frac{1}{2\mathrm{j}}}{s-\mathrm{j}\omega}$$

而

$$L^{-1}\left[\frac{1}{s\pm\mathrm{j}\omega}\right]=\mathrm{e}^{-\mathrm{j}(\pm\omega T)}$$

所以

$$F(z)=Z\left[\frac{\omega}{s^2+\omega^2}\right]=-\frac{1}{2\mathrm{j}}\frac{1}{1-\mathrm{e}^{-\mathrm{j}\omega T}z^{-1}}+\frac{1}{2\mathrm{j}}\frac{1}{1-\mathrm{e}^{\mathrm{j}\omega T}z^{-1}}=\frac{z^{-1}\sin\omega T}{1-\mathrm{e}^{-\mathrm{j}\omega T}z^{-1}-\mathrm{e}^{\mathrm{j}\omega T}z^{-1}+z^{-2}}$$

$$=\frac{z^{-1}\sin\omega T}{1-2z^{-1}\cos\omega T+z^{-2}}=\frac{z\sin\omega T}{z^2-2z\cos\omega T+1}$$

像拉普拉斯变换一样，可以列出常用简单函数的 Z 变换，在解离散问题时可以直接查表。常用函数的拉普拉斯变换和 Z 变换如表 5.2 所示，利用此表可以根据给定的函数或其拉普拉斯变换直接查出其对应的 Z 变换，不必进行繁琐的计算，这也是实际中广泛应用的方法。

表 5.2 常用函数的拉普拉斯变换和 Z 变换表

序号	拉普拉斯变换 $F(s)$	时间函数 $f(t)$	Z 变换 $F(z)$
1	1	$\delta(t)$	1
2	e^{-KTs}	$\delta(t-kT)$	z^{-k}
3	$\frac{1}{1-\mathrm{e}^{-Ts}}$	$\delta_T(t)=\sum_{n=0}^{\infty}\delta(t-nT)$	$\frac{z}{z-1}$
4	$\frac{1}{s}$	$1(t)$	$\frac{z}{z-1}$
5	$\frac{1}{s^2}$	t	$\frac{Tz}{(z-1)^2}$
6	$\frac{1}{s^3}$	$\frac{t^2}{2}$	$\frac{T^2z(z+1)}{2\,(z-1)^3}$
7	$\frac{1}{s^{n+1}}$	$\frac{t^n}{n!}$	$\lim_{a\to 0}\frac{(-1)^n}{n!}\frac{\partial^n}{\partial a^n}\left(\frac{z}{z-\mathrm{e}^{-aT}}\right)$
8	$\frac{1}{s+a}$	e^{-aT}	$\frac{z}{z-\mathrm{e}^{-aT}}$
9	$\frac{1}{(s+a)^2}$	$t\mathrm{e}^{-aT}$	$\frac{Tz\mathrm{e}^{-aT}}{(z-\mathrm{e}^{-aT})^2}$
10	$\frac{a}{s(s+a)}$	$1-\mathrm{e}^{-aT}$	$\frac{(1-\mathrm{e}^{-aT})z}{(z-1)(z-\mathrm{e}^{-aT})}$
11	$\frac{a-b}{(s+a)(s+b)}$	$\mathrm{e}^{-bT}-\mathrm{e}^{-aT}$	$\frac{z}{z-\mathrm{e}^{-bT}}-\frac{z}{z-\mathrm{e}^{-aT}}$
12	$\frac{\omega}{s^2+\omega^2}$	$\sin\omega t$	$\frac{z\sin\omega T}{z^2-2z\cos\omega T+1}$

续表

序号	拉普拉斯变换 $F(s)$	时间函数 $f(t)$	Z 变换 $F(z)$
13	$\frac{s}{s^2+\omega^2}$	$\cos\omega t$	$\frac{z(z-\cos\omega T)}{z^2-2z\cos\omega T+1}$
14	$\frac{\omega}{(s+a)^2+\omega^2}$	$e^{-aT}\sin\omega t$	$\frac{ze^{-aT}\sin\omega T}{z^2-2ze^{-aT}\cos\omega T+e^{-2aT}}$
15	$\frac{s+a}{(s+a)^2+\omega^2}$	$e^{-aT}\cos\omega t$	$\frac{z^2-ze^{-aT}\cos\omega T}{z^2-2ze^{-aT}\cos\omega T+e^{-2aT}}$
16	$\frac{1}{s-(1/T)\ln a}$	$a^{t/T}$	$\frac{z}{z-a}$

3) Z 变换的性质和定理

在 Z 变换中有一些与拉普拉斯变换类似的性质和定理，这些性质和定理对扩大 Z 变换应用有着重要作用，应用这些定理可使 Z 变换的运算变得简单方便，并可用于计算或直接分析离散控制系统。其中 Z 变换最基本的性质和定理如下（这里假设 $f(t)$ 的 Z 变换 $F(z)$ 存在，并且对于 $t<0$ 时 $f(t)=0$）。

(1) 线性定理

由 Z 变换的定义可知，Z 变换是线性变换，若 $F_1(z)=Z[f_1(t)]$，$F_2(z)=Z[f_2(t)]$，且 a 和 b 为任意常数，则有

$$Z[af_1(t)+bf_2(t)]=aF_1(z)+bF_2(z) \tag{5.8}$$

(2) 迟后定理

若连续函数 $f(t)$ 的 Z 变换为 $F(z)=Z[f(t)]$，并且当 $t<0$ 时有 $f(t)=0$，则有

$$Z[f(t-kT)]=z^{-k}F(z) \tag{5.9}$$

其中 k 为零或正整数。

(3) 超前定理

若连续函数 $f(t)$ 的 Z 变换为 $F(z)=Z[f(t)]$，并且当 $t<0$ 时有 $f(t)=0$，则有

$$Z[f(t+kT)]=z^kF(z)-z^k\sum_{j=0}^{k-1}f(jT)z^{-j} \tag{5.10}$$

其中 k 为零或正整数。

(4) 复位移定理

如果 $f(t)$ 的 Z 变换为 $F(z)=Z[f(t)]$，则有

$$Z[e^{\pm aT}f(t)]=F(e^{\mp aT}z) \tag{5.11}$$

(5) 初值定理

如果 $f(t)$ 的 Z 变换为 $F(z)=Z[f(t)]$，且 $\lim\limits_{z\to\infty}F(z)$ 存在，则 $f(t)$ 或 $f(k)$ 的初值 $f(0)$ 为

$$f(0)=\lim_{z\to\infty}F(z) \tag{5.12}$$

(6) 终值定理

如果 $f(t)$ 的 Z 变换为 $F(z)=Z[f(t)]$，且 $(z-1)F(z)$ 对 z 的所有导数都存在，则有

$$\lim_{t\to\infty}f(t)=\lim_{z\to1}(z-1)F(z)=\lim_{z\to1}(1-z^{-1})F(z) \tag{5.13}$$

(7) 卷积定理

如果 $f_1(t)$ 和 $f_2(t)$ 的 Z 变换为 $F_1(z)=Z[f_1(t)]$，$F_2(z)=Z[f_2(t)]$，则有

$$Z[f_1(t)^* f_2(t)] = Z\left[\sum_{i=0}^{k} f_1(iT)f_2(kT-iT)\right] = F_1(z)F_2(z) \tag{5.14}$$

5.1.2 Z 反变换

在连续系统设计中，通过拉普拉斯变换把描述系统的微分方程转换为 s 的代数方程，然后写出系统的传递函数，即可用拉普拉斯变换法求出系统的时间响应，从而简化对连续系统的分析研究。同样，在离散系统中进行 Z 变换，是为了把描述离散系统的差分方程转换成 Z 的代数方程，然后写出离散系统的脉冲传递函数，再用 Z 反变换的方法求出离散系统的时间响应。

与 Z 变换相反，由象函数 $F(z)$求取相应的离散时间原函数 $f^*(t)$的运算，就是所谓的 Z 反变换，有时也叫逆 Z 变换，记为

$$f^*(t) = Z^{-1}[F(z)] \tag{5.15}$$

求 Z 反变换的方法很多，比如查表法、利用 Z 变换的性质求反变换、幂级数展开法、部分分式展开法和留数法等。对于常见的典型信号，其 Z 反变换可以通过查表 5.1 中的 Z 变换表得到，对于一些表中无法直接查询的，可以通过幂级数展开法、部分分式法和留数计算法来求得。下面重点介绍这三种 Z 反变换的求解方法。

1. 幂级数展开法

幂级数展开法也称为长除法，这种方法是由函数的 Z 变换表达式，利用长除法求出按 Z^{-k}升幂排列的级数展开式，再经拉普拉斯反变换，即可求出原函数的脉冲序列。

根据 Z 变换的定义，将式(5.6)展开成 Z^{-1}的无穷幂级数，可知

$$F(z) = \sum_{k=0}^{+\infty} f(kT)z^{-k} = f(0) + f(T)z^{-1} + f(2T)z^{-2} + \cdots + f(kT)z^{-k} + \cdots \tag{5.16}$$

由上式可知，只要用某种方法将要作 Z 反变换的 $F(z)$展开成 z^{-1}的幂级数形式，即可对应获得 $F(z)$相应的时间序列 $f(kT)$。

设象函数 $F(z)$是 Z 的有理函数，可以表示为两个 Z 的多项式之比，即

$$F(z) = \frac{b_0 z^m + b_1 z^{m-1} + \cdots + b_m}{a_0 z^n + a_1 z^{n-1} + \cdots + a_n}(n \geqslant m) \tag{5.17}$$

对式(5.17)用长除法，即直接用分子多项式除以分母多项式，所得商展开成 Z^{-1}升幂排列的级数形式，即

$$F(z) = c_0 z^0 + c_1 z^{-1} + c_2 z^{-2} + \cdots \tag{5.18}$$

比较式(5.18)和式(5.16)可知其具有相同的形式，而式(5.16)是 Z 变换的定义式，因此式(5.18)的系数 $c_n(n=0,1,2,\cdots)$就是 $f(t)$在采样时刻的值 $f(kT)$，故将 $z=e^{Ts}$代入式(5.18)再求取其拉普拉斯反变换，则得到对应的原函数，即

$$f(kT) = c_0\delta(t) + c_1\delta(t-T) + c_2\delta(t-2T) + \cdots \tag{5.19}$$

【例 5.5】 已知象函数 $F(z)=\dfrac{z}{(z+1)(z+2)}$，试求其原函数 $f^*(t)$。

解：首先将 $F(z)$写出 Z 多项式之比的形式，$F(z)=\dfrac{z}{(z+1)(z+2)}=\dfrac{z}{z^2+3z+2}$应用长

除法将象函数 $F(z)$ 展开，即

$$
\begin{array}{r|l}
 & z^{-1}-3z^{-2}+7z^{-3}-15z^{-4}+\cdots \\
\hline
z^2+3z+2 & z \\
 & \underline{z\quad +3\quad +2z^{-1}} \\
 & \quad -3\quad -2z^{-1} \\
 & \quad \underline{-3\quad -9z^{-1}-6z^{-2}} \\
 & \qquad\qquad 7z^{-1}+6z^{-2} \\
 & \qquad\qquad \underline{7z^{-1}+21z^{-2}+14z^{-3}} \\
 & \qquad\qquad\qquad -15z^{-2}-14z^{-3} \\
 & \qquad\qquad\qquad \underline{-15z^{-2}-45z^{-3}-30z^{-4}} \\
 & \qquad\qquad\qquad\qquad\quad 31z^{-3}+30z^{-4} \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \vdots
\end{array}
$$

由长除法结果可知

$$F(z)=z^{-1}-3z^{-2}+7z^{-3}-15z^{-4}+\cdots$$

故可得其原函数

$$f(kT)=\delta(t-T)-3\delta(t-2T)+7\delta(t-3T)-15\delta(t-4T)+\cdots$$

由例 5.5 可以看出，采用幂级数展开法，用手工就可以求出 $f^*(t)$ 的前几项，数学难度低，在分析实际问题时，如果只需求得序列的初段，则应用此方法还是很有意义的。但是一般来说，幂级数展开法得不出 $f(kT)$ 的封闭表达式。

2. 部分分式法

部分分式展开法又称为查表法，其基本思想是基于 Z 变换的线性性质，将 $F(z)$ 展开成若干个简单基本分式和的形式，然后通过查 Z 变换表，或者利用熟知的一些基本对应关系求得 $f(kT)$，这种方法可以求出脉冲序列函数的封闭形式，具体方法和求拉普拉斯反变换的部分分式展开法相似，其步骤如下。

(1) 首先将 $F(z)$ 写成有理式的标准形式，即

$$F(z)=\frac{N(z)}{D(z)}=\frac{b_m z^m+b_{m-1}z^{m-1}+\cdots+b_1 z+b_0}{a_n z^n+a_{n-1}z^{n-1}+\cdots+a_1 z+a_0}\quad (n\geqslant m)$$

(2) 因各种基本初等函数的 Z 变换分子都带有因子 z，所以，需要将 $F(z)$ 除以 z，得到 $\frac{F(z)}{z}$。

(3) 将 $\frac{F(z)}{z}$ 分解为部分分式，方法和拉普拉斯变换的一样，也分为特征方程无重根和有重根两种情况。

(4) 将展开的部分分式各项乘以因子 z，即得到 $F(z)$ 的表达式。

(5) 查表，对各部分分式进行 Z 反变换。

(6) 写出原函数 $f(kT)$。

下面分两种情况来讨论部分分式展开法的 Z 反变换。

1) 特征方程无重根的情况

特征方程无重根的情况，也即 $F(z)$ 的所有极点两两是互不相同的，没有重极点。由于指数函数 e^{-aT} 的 Z 变换为 $\frac{z}{z-e^{-aT}}$，而 $F(z)$ 的部分分式可以展开为 $\frac{A_i}{z-p_i}$ 形式的诸项和，所

以先求出$\frac{F(z)}{z}$的部分分式，即

$$\frac{F(z)}{z}=\frac{A_0}{z}+\frac{A_1}{z-z_1}+\cdots+\frac{A_n}{z-z_n}=\sum_{i=0}^{n}\frac{A_i}{z-z_i} \tag{5.20}$$

再由上式写出 $F(z)$的部分分式之和

$$F(z)=A_0+\sum_{i=1}^{n}\frac{A_iz}{z-z_i} \tag{5.21}$$

然后逐项通过查 Z 变换表，即可得

$$f(kT)=A_0\delta(t)+\sum_{i=0}^{n}A_iz_i^k \tag{5.22}$$

【例 5.6】 已知象函数 $F(z)=\frac{z}{(z+1)(z+2)}$，试求其原函数 $f^*(t)$。

解：首先将 $F(z)$展开，即

$$F(z)=\frac{z}{(z+1)(z+2)}=\frac{z}{z+1}-\frac{z}{z+2}$$

查表可知

$$Z^{-1}\left[\frac{z}{z-a}\right]=a^k$$

故可得其原函数为：

$$f(kT)=(-1)^k-(-2)^k=\delta(t-T)-3\delta(t-2T)+7\delta(t-3T)-15\delta(t-4T)+\cdots$$

从例 5.5 和例 5.6 可知，两种计算方法求得的结果是一样的。

【例 5.7】 已知 Z 变换象函数 $F(z)=\frac{3z}{(z-1)(z+2)}$，试求其原函数 $f^*(t)$。

解：由于

$$\frac{F(z)}{z}=\frac{1}{z-1}-\frac{1}{z+2}$$

上式两边同时乘以 z，得

$$F(z)=\frac{z}{z-1}-\frac{z}{z+2}$$

查表可知

$$Z^{-1}\left[\frac{z}{z-a}\right]=a^k$$

故可得其原函数为：

$$f(kT)=(1)^k-(-2)^k$$

2）特征方程有重根的情况

特征方程有重根的情况，也即 $F(z)$的所有极点中有相同的情况，有重极点。首先求出$\frac{F(z)}{z}$的部分分式，即

$$\frac{F(z)}{z}=\frac{N(z)}{z(z-z_1)^m}=\frac{K_0}{z}+\frac{K_{11}}{(z-z_1)^m}+\cdots+\frac{K_{1m}}{z-z_1}\quad（假设有\ m\ 重根） \tag{5.23}$$

其中

$$K_{1i}=\frac{1}{(i-1)!}\cdot\frac{\mathrm{d}^{i-1}}{\mathrm{d}z^{i-1}}\left[(z-z_1)^m\frac{F(z)}{z}\right]\bigg|_{z=z_1}$$

算出各分式的待定系数，然后通过诸项查 Z 变换表即可求得。

【例 5.8】 已知象函数 $F(z)=\dfrac{2z^3-40z}{(z-2)^3(z-4)}$，试求其反变换。

解：通过 $F(z)$ 可知 $z=2$ 是三阶极点，$z=4$ 是一阶极点，首先将 $\dfrac{F(z)}{z}$ 展开为部分分式形式，即

$$\frac{F(z)}{z}=\frac{2z^2-40}{(z-2)^3(z-4)}=\frac{K_{11}}{(z-2)^3}+\frac{K_{12}}{(z-2)^2}+\frac{K_{13}}{(z-2)}+\frac{K_2}{z-4}$$

其中

$$K_{11}=\left[(z-2)^3\frac{F(z)}{z}\right]\Bigg|_{z=2}=16$$

$$K_{12}=\frac{1}{(2-1)!}\cdot\frac{\mathrm{d}}{\mathrm{d}z}\left[(z-2)^3\frac{F(z)}{z}\right]\Bigg|_{z=2}=4$$

$$K_{13}=\frac{1}{(3-1)!}\cdot\frac{\mathrm{d}^2}{\mathrm{d}z^2}\left[(z-2)^3\frac{F(z)}{z}\right]\Bigg|_{z=2}=1$$

$$K_2=\left[(z-4)\frac{F(z)}{z}\right]\Bigg|_{z=4}=-1$$

所以

$$\frac{F(z)}{z}=\frac{16}{(z-2)^3}+\frac{4}{(z-2)^2}+\frac{1}{z-2}+\frac{-1}{z-4}$$

即

$$F(z)=\frac{16z}{(z-2)^3}+\frac{4z}{(z-2)^2}+\frac{z}{z-2}+\frac{-z}{z-4}$$

查表可知

$$Z^{-1}\left[\frac{z}{z-a}\right]=a^k$$

同时由于

$$Z^{-1}\left[\frac{z^3}{(z-a)^3}\right]=\frac{(k+1)(k+2)}{2}a^k,\quad Z^{-1}\left[\frac{z^2}{(z-a)^2}\right]=(k+1)a^k$$

（这两个函数的 Z 反变换可以通过留数计算法推导出来）利用 Z 变换性质的迟后定理可知

$$Z^{-1}\left[\frac{z}{(z-2)^3}\right]=\frac{(k+1-2)(k+2-2)}{2}2^{k-2}=\frac{k(k-1)}{8}2^k=\frac{k(k-1)}{8}2^k$$

$$Z^{-1}\left[\frac{z}{(z-2)^2}\right]=(k+1-1)2^{k-1}=k\cdot 2^{k-1}=k\cdot 2^{k-1}$$

利用上述分式的反变换结果可得其原函数为：

$$f(k)=16\cdot\frac{k(k-1)}{8}\cdot 2^k+4k\cdot 2^{k-1}+2^k-4^k=-4^k+2^k+k^2\cdot 2^{k+1}$$

3）留数计算法（围线积分法）

由 Z 变换的定义可知

$$F(Z)=\sum_{k=0}^{+\infty}f(kT)z^{-k}=f(0)+f(T)z^{-1}+f(2T)z^{-2}+\cdots+f(kT)z^{-k}+\cdots$$

将上式两端同时乘以 z^{k-1}，得

$$F(z)z^{k-1}=f(0)z^{k-1}+f(T)z^{k-2}+f(2T)z^{k-3}+\cdots+f(kT)z^{-1}+\cdots\tag{5.24}$$

并对上式两边作闭路积分，闭合的积分路径线取以 $z=0$ 为圆心且包围 $F(z)z^{k-1}$ 的全部极点的圆 C，即

$$\oint_C F(z)\cdot z^{k-1}\mathrm{d}z=\oint_C f(0)z^{k-1}\mathrm{d}z+\oint_C f(T)z^{k-2}\mathrm{d}z+\oint_C f(2T)z^{k-3}\mathrm{d}z+\cdots+\oint_C f(kT)z^{-1}\mathrm{d}z+\cdots \tag{5.25}$$

由复变函数的柯西定理可知

$$\oint_C f(kT)z^{-1}\mathrm{d}z=2\pi\mathrm{j}\cdot f(kT) \tag{5.26}$$

其余各项积分为零，所以有

$$\oint_C F(z)z^{k-1}\mathrm{d}z=2\pi\mathrm{j}\cdot f(kT) \tag{5.27}$$

也即

$$f(kT)=\frac{1}{2\pi\mathrm{j}}\oint_C F(z)z^{k-1}\mathrm{d}z \tag{5.28}$$

根据留数定理，上式可表示为

$$f(kT)=\frac{1}{2\pi\mathrm{j}}\oint_C F(z)z^{k-1}\mathrm{d}z=\sum_{m=1}^{n}\mathrm{Res}[F(z)z^{k-1}]\Big|_{z=z_m} \tag{5.29}$$

其中，z_m 为 $F(z)z^{k-1}$ 的第 m 个极点，Res[]表示极点处的留数，n 为 $F(z)z^{k-1}$ 的极点数。上式也表明 $f(kT)$ 等于函数 $F(z)z^{k-1}$ 在其全部极点上的留数之和，这也是留数计算法求 Z 反变换的方法。

在计算留数时，如果 $F(z)z^{k-1}$ 含有简单的单阶极点 z_m，则相应的留数为

$$\mathrm{Res}[F(z)z^{k-1},z_m]=\lim_{z\to z_m}[(z-z_m)F(z)z^{k-1}] \tag{5.30}$$

如果 $F(z)z^{k-1}$ 含有的 r 重阶极点 z_m，则该极点的留数为

$$\mathrm{Res}[F(z)z^{k-1},z_m]=\frac{1}{(r-1)!}\lim_{z\to z_m}\frac{\mathrm{d}^{r-1}}{\mathrm{d}z^{r-1}}[(z-z_m)^r F(z)z^{k-1}] \tag{5.31}$$

【例 5.9】 已知象函数 $F(z)=\dfrac{z}{(z+1)(z+2)}$，试用留数法求其 Z 反变换。

解：因为 $F(z)$ 有两个极点：-1 和 -2，先求出 $F(z)z^{k-1}$ 对这两个极点的留数。

$$\mathrm{Res}\left[\frac{zz^{k-1}}{(z+1)(z+2)},-1\right]=\lim_{z\to -1}\frac{\mathrm{d}}{\mathrm{d}s}\left[(z+1)\frac{zz^{k-1}}{(z+1)(z+2)}\right]=(-1)^k$$

$$\mathrm{Res}\left[\frac{zz^{k-1}}{(z+1)(z+2)},-2\right]=\lim_{z\to -2}\frac{\mathrm{d}}{\mathrm{d}s}\left[(z+2)\frac{zz^{k-1}}{(z+1)(z+2)}\right]=-(-2)^k$$

故由留数计算法可得其原函数为：

$$f(kT)=(-1)^k-(-2)^k=\delta(t-T)-3\delta(t-2T)+7\delta(t-3T)-15\delta(t-4T)+\cdots$$

从例 5.5、例 5.6 和例 5.9 可知，三种计算方法求得的结果是一样的。

5.2 脉冲传递函数和差分方程

在连续系统中一般用传递函数来表示系统的动态特性，它是研究系统性能的重要基础。在离散系统中，也存在着和连续系统相对应的数学描述形式，通常有差分方程、脉冲传递函数、单位脉冲响应序列和离散状态空间这四种数学描述形式。在描述离散控制系统输入和

输出信号间的关系时，主要采用脉冲传递函数，有时也简称为 Z 传递函数，其重要性和用传递函数来描述连续控制系统是一样的。

5.2.1 离散系统

在控制系统中的各个变量，比如输入量、输出量和偏差量等都是时间 t 的连续函数的系统称做连续控制系统；而离散系统简单地说就是其输入量和输出量均为离散信号的物理系统。随着脉冲技术、数字计算机特别是微处理器技术的发展，系统中存在的信号不是连续的，而是用离散的脉冲序列或数字序列来表示，因此离散系统和连续系统之间存在着本质的差别，不能用有关的连续系统理论直接来分析离散系统。

离散系统在数学上可以简单地抽象为一种由系统的离散输入信号 $r(k)$ 到系统的离散输出 $y(k)$ 的数学变换或者映射，如果将这种变换或映射以符号 $f[\,]$ 表示，则离散系统可表示为 $y(k)=f[r(k)]$。

如果离散系统的输入信号到输出信号之间的关系满足叠加定理，则称为线性离散系统，即当 $y_1(k)=f[r_1(k)]$，$y_2(k)=f[r_2(k)]$，输入信号为 $r(k)=ar_1(k)+br_2(k)$ 时（其中 a，b 为任意常数），则系统相应的输出信号可表示为

$$y(k) = f[ar_1(k)+br_2(k)] = af[r_1(k)]+bf[r_2(k)] = ay_1(k)+by_2(k)$$

如果不满足叠加定理，系统就是非线性离散系统。同样与连续系统一样，离散系统还可以分为时变系统和时不变系统，时不变系统就是指输入信号和输出信号之间的关系是不随时间变化的，即若系统的输出 $y(k)=f[r(k)]$，则当系统的输入为 $r(k-n)$ 时，则其输出响应为 $y(k-n)=f[r(k-n)]$，时不变系统又称为定常离散系统或位移不变系统。

如果离散系统的输入信号和输出信号之间的关系既满足叠加定理，同时其之间的变化关系又不随时间变化而变化，则称为线性时不变离散系统。工程中的大多数计算机控制系统可以近似为线性时不变离散系统来处理。

5.2.2 差分方程

1. 线性常系数差分方程

对于线性连续系统其输入和输出之间可用线性常系数微分方程来描述，与线性连续系统类似，对于线性离散系统，其输入和输出之间可用线性常系数差分方程来描述。常系数差分方程的基本形式和常系数微分方程类似，但差分方程有前向差分方程和后向差分方程之分。

两个采样点信息之间的差值即称为差分，在差分方程中，自变量是离散的，方程的各项包含有这种离散变量的函数，还包含有这种函数增序或减序的函数。差分方程未知函数中变量的最高和最低序号的差数称为方程的阶数。对于一个单输入单输出线性定长离散系统，在某一个采样时刻的输出值 $y(k)$ 不仅与这一时刻的输入值 $r(k)$ 有关，而且与过去时刻的输入值 $r(k-1)$，$r(k-2)$，…，有关，还与过去的输出值 $y(k-1)$，$y(k-2)$，…，有关。这种线性离散系统的差分方程一般式为

$$\begin{aligned} &y(k)+a_1y(k-1)+a_2y(k-2)+\cdots+a_ny(k-n) \\ &= b_0r(k)+b_1r(k-1)+b_2r(k-2)+\cdots+b_mr(k-m) \end{aligned} \tag{5.32}$$

式中，系数 a_1，a_2，…，a_n，b_0，b_1，b_2，…，b_m 均为常实数，n 为方程的阶次。因此式(5.32)称为 n 阶后向非齐次差分方程。对于 n 阶差分方程，系数 $a_n\neq 0$，其余系数 a_1，a_2，…，a_{n-1} 都可能

为零。若 $a_n=0$,相当于方程的阶次降为 $n-1$ 阶。与式(5.32)类似,n 阶前向非齐次差分方程的基本形式为

$$\begin{aligned} &y(k+n)+a_1y(k+n-1)+a_2y(k+n-2)+\cdots+a_ny(k) \\ &=b_0r(k+m)+b_1r(k+m-1)+b_2r(k+m-2)+\cdots+b_mr(k) \end{aligned} \tag{5.33}$$

或

$$y(k+n)=-\sum_{j=1}^{n}a_jy(k+n-j)+\sum_{j=0}^{m}b_jr(k+m-j) \tag{5.34}$$

式中,系数 $a_1,a_2,\cdots,a_n,b_0,b_1,b_2,\cdots,b_m$ 均为常实数,n 为方程的阶次。

对于实际存在的有因果关系的物理系统,式(5.32)和式(5.33)的方程中总是 $m\leqslant n$,如果 $m>n$ 则表明方程描述的离散系统的输出信号超前于输入信号,即还没有输入信号作用于系统时,其系统已经有对应的输出信号了,或者说系统当前时刻的输出 $y(k)$与未来时刻的输入量 $x(k+j)(j>0)$有关,这种情况在现实的物理系统中是矛盾的,不可能出现的。当 $m<n$ 表明系统存在着相应的延迟,若 $n-m=d$,则相应的离散系统输出相对于输入有 d 个采样周期的延迟。

2. 差分方程的求解

差分方程的求解就是在系统初始值和输入序列已知的条件下,求出差分方程描述的系统在任何时刻的输出序列值。常系数线性差分方程的常用求解方法有经典解法、基于解析方法的 Z 变换法和基于计算机求解的迭代法三种基本方法。下面重点介绍 Z 变换法求解差分方程和迭代法。

(1) Z 变换法求解差分方程的步骤是:对描述离散系统的差分方程进行 Z 变换,并利用 Z 变换的实数位移定理,将时域差分方程化为 z 域的代数方程,求其解,再将 z 域的代数方程经 Z 反变换求得差分方程的时域解。

【例 5.10】 用 Z 变换求解下列差分方程。

$$y(k+2)+5y(k+1)+6y(k)=0$$

初始条件为 $y(0)=0,y(1)=1$。

解:由超前定理对差分方程两边求 Z 变换,得

$$[z^2y(z)-z^2y(0)-zy(1)]+[5zy(z)-5zy(0)]+6y(z)=0$$

将初始条件代入上式,并化简解得

$$(z^2+5z+6)y(z)=z$$

整理得

$$y(z)=\frac{z}{z^2+5z+6}=\frac{z}{(z+2)(z+3)}=\frac{z}{z+2}-\frac{z}{z+3}$$

对上式两边进行 Z 反变换,有

$$y(k)=(-2)^k-(-3)^k \quad (k=0,1,2,\cdots)$$

(2) 迭代法是已知离散系统的差分方程和输入序列,并且给定输出序列的初始值时,就可以利用递推迭代关系逐步计算出所需要的输出序列值的方法。

【例 5.11】 已知一个离散系统的差分方程为

$$y(k+2)+5y(k+1)+6y(k)=0$$

初始条件为 $y(0)=0,y(1)=1$,试用迭代法求解差分方程。

解:将原差分方程改写成迭代式:$y(k+2)=-5y(k+1)-6y(k)$

考虑到初始条件,逐步以 $k=0,1,2,\cdots$,代入差分方程,一步一步迭代求解如下:

当 $k=0$ 时, $y(2)=-5y(1)-6y(0)=-5\times1-6\times0=-5$;

当 $k=1$ 时, $y(3)=-5y(2)-6y(1)=-5\times(-5)-6\times1=19$;

当 $k=2$ 时, $y(4)=-5y(3)-6y(2)=-5\times19-6\times(-5)=-65$;

当 $k=3$ 时, $y(5)=-5y(4)-6y(3)=-5\times(-65)-6\times19=211$;

当 $k=4$ 时, $y(6)=-5y(5)-6y(4)=-5\times211-6\times(-65)=-665$;

$\vdots$

而在例 5.10 中,已求得 $y(k)=(-2)^k-(-3)^k(k=0,1,2,\cdots)$,也即

$$y(1)=(-2)^1-(-3)^1=1$$

$$y(2)=(-2)^2-(-3)^2=-5$$

$$y(3)=(-2)^3-(-3)^3=19$$

$$y(4)=(-2)^4-(-3)^4=-65$$

$$y(5)=(-2)^5-(-3)^5=211$$

$$y(6)=(-2)^6-(-3)^6=-665$$

显然通过迭代法可以得到任意 kT 时刻的输出序列 $y(kT)$的值,与 Z 变换法求解求得输出序列的结果是一致的。但求出的不是 $y(kT)$的数学解析表达式。迭代法的优点是便于用计算机编程来求解实现。比如针对例 5.11,利用 MATLAB 语言编程如下。

```
% 由于在 MATLAB 中数组的编号必须从 1 开始,所以先必须给 y(0)赋值,根据初始条件写作 y(0) = 0,
并根据编程循环的需要计算 y(2) = - 5
y(0) = 0; y(1) = 1; y(2) = - 5;
for   k = 1: 4                          % 循环次数即为要计算的 y(k)的拍数,根据需要来设置
    y(k + 2) = - 5 * y(k + 1) - 6 * y(k)  % 用迭代法计算 y(k)的值
end
计算的结果如下: y =    1     - 5     19    - 65    211   - 665
```

与例 5.10 和例 5.11 的结果是一致的。

差分方程的解,可以提供线性定常离散系统在给定输入序列作用下的输出序列响应特性,但不便于研究系统参数变化对离散系统性能的影响。所以需要引入脉冲传递函数。

5.2.3 脉冲传递函数

1. 脉冲传递函数定义

在连续系统理论中,传递函数的定义为在初始静止的状态下,即系统的初始值为零,系统输出信号的拉普拉斯变换 $Y(s)$与输入信号的拉普拉斯变换 $R(s)$之比,即

$$G(s)=\frac{Y(s)}{R(s)} \tag{5.35}$$

类似地,线性离散控制系统的传递函数可定义为:在零初始条件下,一个系统或环节的输出脉冲序列的 Z 变换 $Y(z)$与输入脉冲序列的 Z 变换 $R(z)$之比,即

$$G(z)=\frac{Y(z)}{R(z)} \tag{5.36}$$

为了区别于连续系统,线性离散控制系统的传递函数称为脉冲传递函数或 z 传递函数。零初始条件是指在 $t<0$ 时,输入脉冲序列各采样值 $R(-T)$、$R(-2T)$、$\cdots$以及输出脉冲序列各采样值 $y(-T)$、$y(-2T)$、$\cdots$均为 0。

根据式(5.36)可知，系统输出的离散信号的 Z 变换可表示为系统的脉冲传递函数与输入信号的 Z 变换的乘积，即

$$Y(z) = G(z) * R(z) \tag{5.37}$$

通过对 $Y(z)$ 的 Z 反变换可求出系统的输出响应 $y(kT)$，即

$$y^*(t) = Z^{-1}[Y(z)] = Z^{-1}[G(z) \cdot R(z)] \tag{5.38}$$

脉冲传递函数是离散系统动态特性的一种数学描述形式，可以用来研究离散系统的性能。与连续系统的传递函数一样，它也只与系统本身的特性有关，而与系统外部输入的信号形式无关，并且仅适用于线性、时不变系统，而不适用于非线性系统和时变系统。

脉冲传递函数反映的是系统输入采样信号与输出采样信号之间的传递关系。但实际上对大多数实际系统来说，其输出信号往往是连续信号 $y(t)$，而不是离散信号 $y^*(t)$，在这种情况下，为了应用脉冲传递函数的概念，可以在系统输出端虚设一个理想采样开关，如图 5.1 中的虚线所示，这一虚设的采样开关与输入端的采样开关同步工作，因此具有相同的采样周期 T。如果系统的实际输出 $y(t)$ 比较平滑，且采样频率较高，则可由 $y^*(t)$ 近似描述 $y(t)$，这样输出 $y(t)$ 就可以通过虚设的采样开关得到离散的输出 $y^*(t)$。$y^*(t)$ 并不是实际产生的，在这种情况下，只是用 $y^*(t)$ 来描述 $y(t)$，可得出系统的脉冲传递函数，从而加以分析系统的性能。必须指出，虚设的采样开关是不存在的，它只是表明了脉冲传递函数所能描述的，只是输出连续函数在采样时刻上的离散值 $y^*(t)$。

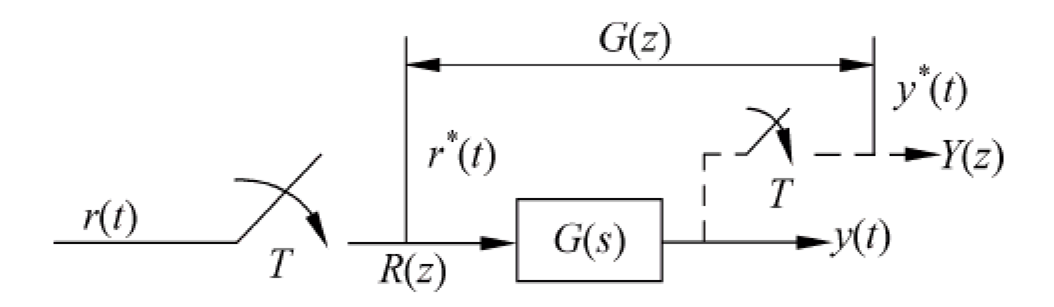


图 5.1　开环离散系统

2. 脉冲传递函数的求解

(1) 由差分方程求脉冲传递函数。脉冲传递函数是离散系统在 z 域的描述形式，而差分方程则是离散系统在时间域的描述形式，若不考虑系统初始值的作用，两者之间是可以互相转换的。

【例 5.12】 已知线性离散系统的差分方程为 $y(k)=y(k-1)+Tr(k-1)$，且初始条件为零，试求系统的脉冲传递函数。

解： 对差分方程两边求 Z 变换，得

$$Y(z)=z^{-1}Y(z)+Tz^{-1}R(z)$$

整理得

$$Y(z)(1-z^{-1})=TR(z)z^{-1}$$

所以系统的脉冲传递函数为

$$G(z) = \frac{Y(z)}{R(z)} = \frac{Tz^{-1}}{1-z^{-1}}$$

(2) 由连续系统传递函数求脉冲传递函数。通过下面的例 5.13 来说明其求解过程。

【例 5.13】 已知某系统的传递函数为 $G(s)=\dfrac{1}{s(s+1)}$，求 $G(s)$ 所对应的脉冲传递函数 $G(z)$。

解： 将 $G(s)$ 分解为部分分式

$$G(s)=\frac{1}{s(s+1)}=\frac{1}{s}-\frac{1}{s+1}$$

查表得出其 Z 变换，即系统脉冲传递函数为

$$G(z)=Z[G(s)]=\frac{z}{z-1}-\frac{z}{z-\mathrm{e}^{-T}}=\frac{z(1-\mathrm{e}^{-T})}{(z-1)(z-\mathrm{e}^{-T})}$$

5.3 数字控制器的连续化设计方法

数字控制器的连续化设计是先根据给定的性能指标以及各项参数，将整个控制系统看成是模拟系统，应用连续系统的设计理论和方法来设计模拟控制器，然后在满足一定条件下，再通过某种离散化近似处理，将模拟控制器离散化为数字控制器，最后通过计算机来实现。

5.3.1 数字控制器连续化设计步骤

数字控制器的连续化设计是先设计出满足系统性能指标要求的模拟控制器 $D(s)$，形成如图 5.2 所示的模拟闭环控制系统，再通过本节介绍的离散化方法将模拟控制器离散化为数字控制器 $D(z)$，这样就得到如图 5.3 所示的计算机控制系统。

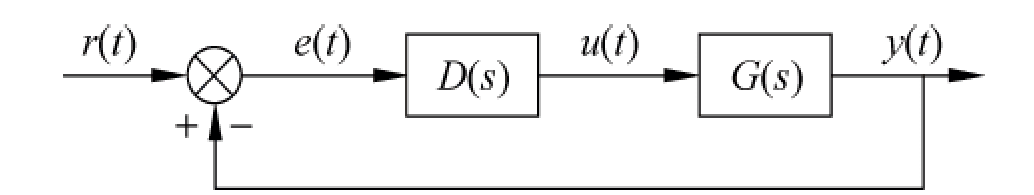


图 5.2 连续控制系统结构图

在如图 5.3 所示的单回路计算机控制系统中，$G(s)$是被控对象的传递函数，$H(s)=\frac{1-\mathrm{e}^{-sT}}{s}$是零阶保持器的传递函数，$D(z)$是数字控制器的脉冲传递函数。现在的设计问题是根据被控对象 $G(s)$，设计出满足系统性能指标的数字控制器 $D(z)$，可按下面的设计步骤进行。

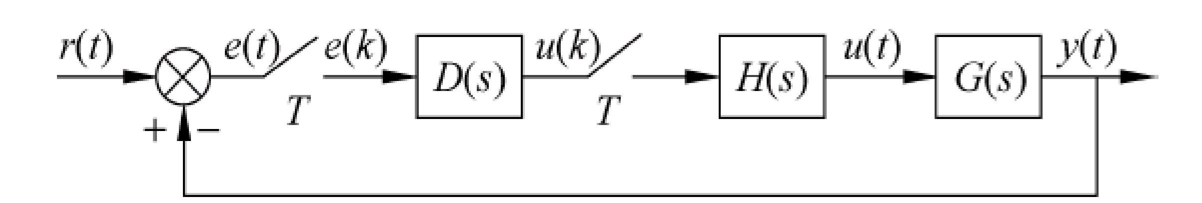


图 5.3 计算机控制系统结构图

1. 设计假想的连续控制器 $D(s)$

根据给定的被控对象的特性以及设计要求的性能指标，来设计如图 5.2 所示的连续控制器 $D(s)$，一种方法是事先确定控制器的结构，比如常用的 PID 算法等，然后通过其控制参数的整定来完成设计；另一种是借助于连续控制系统中比较熟悉的设计方法来设计控制器的结构和参数，比如采用根轨迹法、频率特性法等。

2. 选择合适的采样周期 T

根据香农采样定理，给出了从采样信号中恢复连续信号的最低采样频率$f_s \geqslant 2f_{max}$，也即采样周期 $T\leqslant \pi/\omega_{max}$，理论上应根据采样定理来确定采样周期，但实际上，由于被控对象的物理过程和参数变化比较复杂，致使难以获得模拟信号的最高频率f_{max}。采样定理仅从理论上给出了采样周期的上限，实际采样周期的选择还要受到实际多方面因素的制约。

3. 连续控制器 $D(s)$ 离散化为 $D(z)$

离散化的实质就是选择合理的离散化方法将原连续控制器 $D(s)$ 等效成离散的数字控制器 $D(z)$。"等效"主要是指 $D(s)$ 和 $D(z)$ 在以下表征控制器特性的主要指标方面相近：系统的频带、稳态增益、阶跃响应或脉冲响应形状、频率响应特性等。将连续控制器 $D(s)$ 离散化为数字控制器 $D(z)$ 的方法有很多，比如数值积分法、差分变换法、双线性变换法及修正双线性变换法、零极点匹配法、保持器等价法以及 Z 变换(脉冲响应不变法)等。不同的离散化方法所具有的特性不同，离散化的脉冲传递函数与原传递函数在上述几种特性方面的接近程度是不一致的。因此，应了解不同方法的特点，根据系统性能的要求选择合适的离散化方法。差分变换方法一般有两种，即前向差分法和后向差分法，这里，只重点介绍常用的差分变换法和双线性变换法。

(1) 前向差分法。将 $z=e^{sT}$ 利用级数展开成以下形式

$$z = e^{sT} = 1 + sT + \cdots \approx 1 + sT \tag{5.39}$$

式(5.39)称为前向差分法，并由此解得

$$s = \frac{z-1}{T} \tag{5.40}$$

则 $D(s)$ 离散化的 $D(z)$ 为

$$D(z) = D(s)\big|_{s=\frac{z-1}{T}} \tag{5.41}$$

这也是利用前向差分法求取 $D(z)$ 的计算公式。

(2) 后向差分法。将 $z=e^{sT}$ 变形后再展开成

$$z = e^{sT} = \frac{1}{e^{-sT}} \approx \frac{1}{1-sT} \tag{5.42}$$

式(5.42)称为后向差分法，并由此解得

$$s = \frac{z-1}{Tz} \tag{5.43}$$

则 $D(s)$ 离散化的 $D(z)$ 为

$$D(z) = D(s)\big|_{s=\frac{z-1}{Tz}} \tag{5.44}$$

这也是利用后向差分法求取 $D(z)$ 的计算公式。

(3) 双线性变换法。根据 Z 变换的定义，利用泰勒级数展开可得

$$z = e^{sT} = \frac{e^{\frac{sT}{2}}}{e^{\frac{-sT}{2}}} = \frac{1+\frac{sT}{2}+\cdots}{1-\frac{sT}{2}+\cdots} \approx \frac{1+\frac{sT}{2}}{1-\frac{sT}{2}} \tag{5.45}$$

式(5.45)称为双线性变换或突斯汀(Tustin)变换法，并由此解得

$$s = \frac{2}{T}\frac{z-1}{z+1} \tag{5.46}$$

则 $D(s)$ 离散化的 $D(z)$ 为

$$D(z) = D(s)\big|_{s=\frac{2}{T}\frac{z-1}{z+1}} \tag{5.47}$$

式(5.47)就是利用双线性变换法由 $D(s)$ 求取 $D(z)$ 的计算公式。

对于上述三种变换方法，双线性变换法的优点在于能把 s 左半平面转换到 z 平面的单位圆内，所以一个稳定的连续控制系统变换之后仍将是稳定的，而前向差分法则有可能把一

个稳定的系统变成一个不稳定的离散控制系统，后向差分法也能将一个稳定的系统变换后继续保持稳定。因此在将 $D(s)$离散化 $D(z)$的转换中常用的有后向差分法和双线性变换法，它们也是比较适合工程应用的方法。

4. 数字控制器的计算机算法实现

设计的数字控制器 $D(z)$应变换成易于计算机编程的差分方程的形式。设数字控制器 $D(z)$的一般形式为

$$D(z)=\frac{U(z)}{E(z)}=\frac{b_0+b_1z^{-1}+\cdots+b_mz^{-m}}{1+a_1z^{-1}+\cdots+a_nz^{-n}} \tag{5.48}$$

式中，$m\leqslant n$，各个系数 a_i、b_i 均为实数。

上式(5.48)可改写为

$$U(z)=(-a_1z^{-1}-a_2z^{-2}-\cdots-a_nz^{-n})U(z)+(b_0+b_1z^{-1}+\cdots+b_mz^{-m})E(z) \tag{5.49}$$

上式(5.49)通过 Z 变换的迟后定理用时域可表示为

$$\begin{aligned}u(k)=&-a_1u(k-1)-a_2u(k-2)-\cdots-a_nu(k-n)\\&+b_0e(k)+b_1e(k-1)+\cdots+b_me(k-m)\end{aligned} \tag{5.50}$$

利用式(5.50)即可方便的实现计算机编程，因此式(5.50)称为数字控制器 $D(z)$的控制算法。

5. 校验

控制器 $D(z)$设计完成并得到控制算法后，需要检验其闭环特性是否符合设计要求，可采用数字仿真来验证，如果满足设计要求，则设计结束，否则应重新修改设计。

5.3.2 数字 PID 控制器设计

PID 控制是按偏差的比例、积分和微分进行控制的一种控制规律，PID 是比例(Proportional)、积分(Integral)、微分(Differential)三者的缩写。它具有原理简单、直观易懂、易于实现、参数整定方便、适应性强的优点，是连续系统中技术成熟、应用最广泛的一种控制方式。随着计算机技术的发展，由计算机实现的数字 PID 控制不仅继承了模拟 PID 控制的优点，更是结合了计算机控制的特点，开发出了多种不同形式的 PID 控制算法，使得 PID 控制变得更加灵活多样，并能满足生产过程中的各种需求。

1. 模拟 PID 控制器

在工业控制中，典型的单回路 PID 控制如图 5.4 所示。

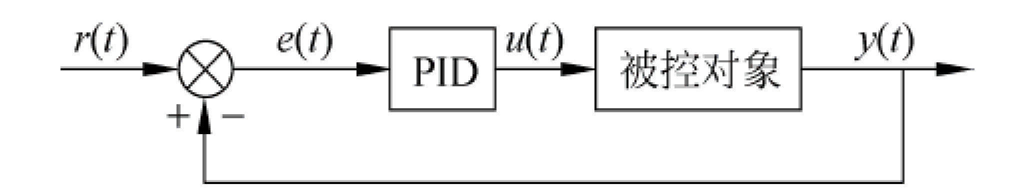


图 5.4 模拟 PID 控制系统图

设系统的误差为 $e(t)$，则模拟 PID 的控制规律为

$$u(t)=K_p\left[e(t)+\frac{1}{T_i}\int_0^t e(t)\mathrm{d}t+T_d\frac{\mathrm{d}e(t)}{\mathrm{d}t}\right] \tag{5.51}$$

式(5.51)中，$u(t)$为调节器的输出信号；$e(t)$为偏差信号，等于给定量和输出量之差；K_p、T_i、T_d 分别为比例系数、积分时间常数和微分时间常数。它所对应的连续时间系统传递函数为：

$$D(s)=\frac{U(s)}{E(s)}=K_p\left(1+\frac{1}{T_i s}+T_d s\right) \tag{5.52}$$

图 5.5 是 PID 控制器的传递函数结构框图，PID 控制器中三项控制作用是互相独立的。比例控制对于误差反应是及时的，在偏差出现的同时，立即就能产生与之成比例的控制作用，能立即减小误差，但缺点是不能消除稳态误差，系统存在余差。积分控制的输出与偏差存在的时间有关，只要偏差存在，积分控制器的输出就会随时间不断地变化，直到消除余差，所以积分作用能够消除静差。但是积分作用缓慢，在偏差刚出现时，积分输出很小，控制作用很弱，因而不能及时克服扰动的影响，使被控量的动态偏差增大，稳定性下降，所以积分控制很少单独使用。微分作用是根据偏差变化的趋势，提前进行控制，从而加快系统的动态响应速度，减小调节时间，因而具有超前控制作用，但微分作用也有可能放大系统的噪声，降低系统的抗干扰能力，并且对于一个固定的偏差，不管其数值多大都不会产生微分作用，因此微分作用也不宜单独使用。

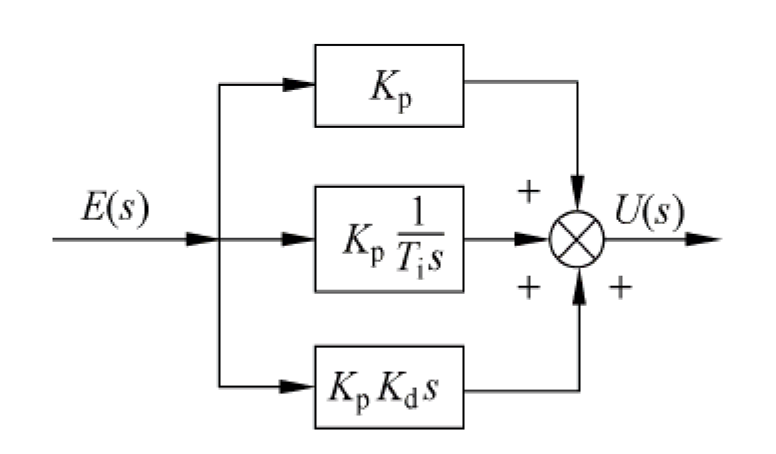


图 5.5　PID 控制器传递函数结构图

PID 控制器的比例、积分和微分作用是通过比例系数、积分时间、微分时间这三个参数来实现的，由于这三个参数兼顾了系统动态过程的现在、过去与将来的信息，使动态过程快速、平稳和准确，所以只要这三个参数选择适当，就可以获得良好的控制品质。在实际工程应用中，可以根据被控对象的特性和要求，灵活地改变其结构，对 PID 三项控制作用进行组合，构成所需要的控制规律，比如有比例控制、比例积分控制和比例微分控制等。

2. 数字 PID 控制器

模拟 PID 控制器是用硬件电路来实现比例、积分和微分控制规律的，而在计算机控制系统中，由于是离散控制，控制器每隔一个控制周期进行一次控制量的计算并输出到执行机构，因此要实现 PID 控制，必须将模拟 PID 算式离散化，数字 PID 控制器是用计算机软件来实现 PID 控制规律的，在计算机控制系统中，PID 控制规律的实现只能用数值逼近的方法。当采样周期足够小时，用求和代替积分，用后向差分代替微分，就可以将模拟 PID 离散化为数字 PID 算法。为此可作如下近似

$$\int_0^t e(t)\mathrm{d}t \approx T\sum_{i=0}^{k} e(i) \tag{5.53}$$

$$\frac{\mathrm{d}e(t)}{\mathrm{d}t} \approx \frac{e(k)-e(k-1)}{T} \tag{5.54}$$

式中，T 为采样周期；k 为采样序号。由式(5.51)、式(5.53)和式(5.54)可得数字 PID 控制算式为

$$u(k)=K_p\left[e(k)+\frac{T}{T_i}\sum_{i=0}^{k}e(i)+T_d\frac{e(k)-e(k-1)}{T}\right] \tag{5.55}$$

式(5.55)中,$e(k)$是第 k 次采样时的偏差值,$e(k-1)$是第 $k-1$ 次采样时的偏差值,并且式(5.55)的控制算法给出了执行机构在采样时刻 k 的位置或控制阀门的开度,所以称为数字 PID 位置型控制算式。

由式(5.55)可以看出,要想计算 $u(k)$,不仅需要偏差信号 $e(k)$和 $e(k-1)$,而且还要把积分项历次的偏差值 $e(i)$进行累加,计算繁琐,占用很大的内存,实际编程使用也不方便。所以在工业控制中常采用另一种称为数字 PID 增量型控制算法的算式。对式(5.55)进行如下的改动即可得到数字 PID 增量型控制算式。

根据推理原理和式(5.55)可写成 $k-1$ 次的位置型 PID 输出表达式,即

$$u(k-1)=K_p\left[e(k-1)+\frac{T}{T_i}\sum_{i=0}^{k-1}e(i)+T_d\,\frac{e(k-1)-e(k-2)}{T}\right] \tag{5.56}$$

用式(5.55)减去式(5.56)可得数字 PID 增量型控制算式为

$$\begin{aligned}\Delta u(k)&=u(k)-u(k-1)\\&=K_P[e(k)-e(k-1)]+K_Ie(k)+K_D[e(k)-2e(k-1)+e(k-2)]\end{aligned} \tag{5.57}$$

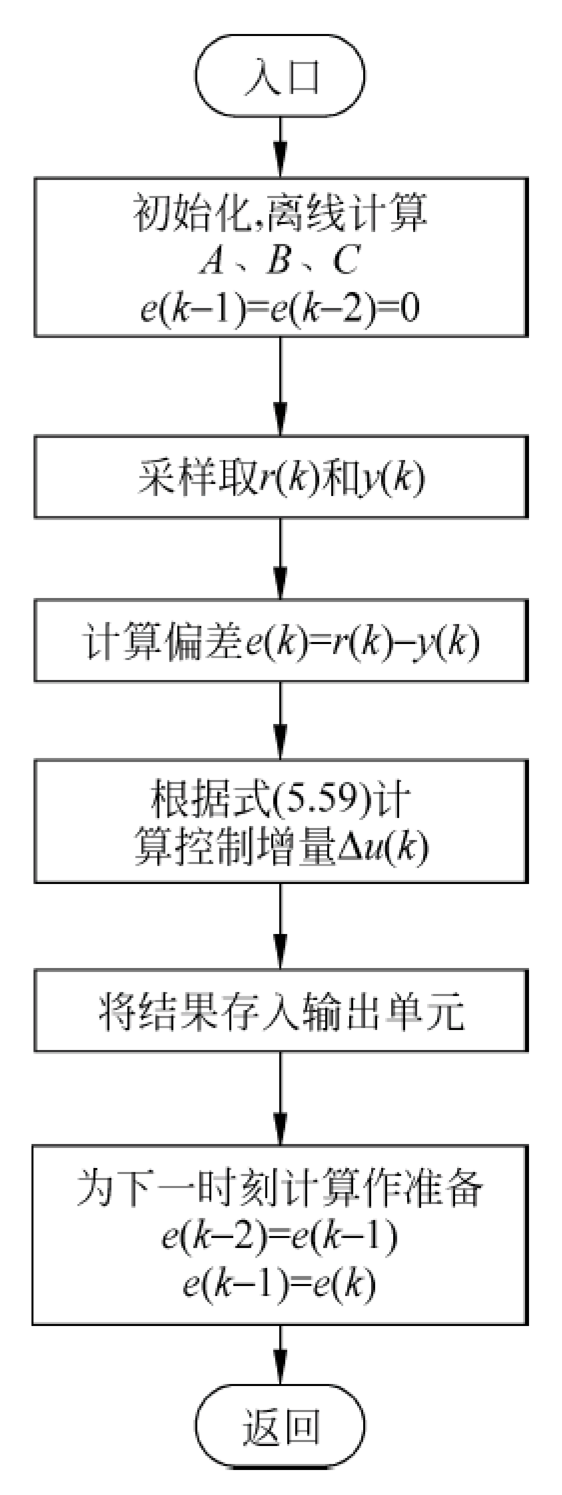


图 5.6 数字 PID 增量型控制算法子程序流程图

式(5.57)中,K_P 为比例增益,与比例带 δ 成倒数关系,即 $K_P=1/\delta$;$K_I=K_PT/T_i$ 为积分系数;$K_D=K_PT_d/T$ 为微分系数。

由式(5.57)可知 k 采样时刻计算机的实际输出控制量为

$$u(k)=u(k-1)+\Delta u(k) \tag{5.58}$$

这也是数字 PID 位置型控制算法的递推算式。

由式(5.57)、式(5.58)可看出,采用增量型 PID 控制算式,要计算第 k 次输出值 $u(k)$,只需知道 $u(k-1)$、$e(k)$、$e(k-1)$和 $e(k-2)$即可,比式(5.55)的计算要简单得多,并且编程简单,数据可以递推使用,占用内存少,运算速度快。为了进一步编程的方便,将式(5.57)可整理成如下形式

$$\Delta u(k)=Ae(k)+Be(k-1)+Ce(k-2) \tag{5.59}$$

式中

$$A=K_p\left(1+\frac{T}{T_i}+\frac{T_d}{T}\right)\quad B=-K_p\left(1+\frac{2T_d}{T}\right)\quad C=K_p\frac{T_d}{T}$$

在编写程序时,可以根据事先确定的比例系数、积分系数和微分系数,计算出 A、B、C,存入内存单元即可。图 5.6 给出了数字 PID 增量型控制算法子程序流程图。

3. 两种标准数字 PID 控制算法比较

数字 PID 控制的位置型和增量型算法本质是相同的,只是形式不同而已,对系统的控制作用,两种算法是完全相同的,如图 5.7 所示。

增量型算法虽然只是在位置型算法上作了一些改动,但它却有不少优点。

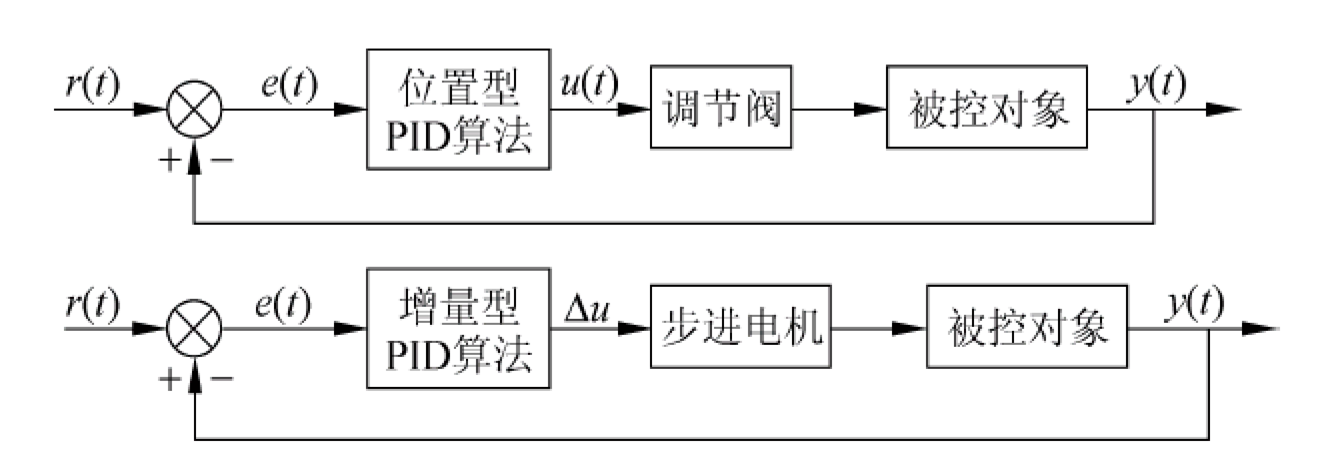


图 5.7 数字 PID 位置型和增量型控制系统示意图

(1) 位置式每次输出与整个过去状态有关,计算式中要用到过去偏差的累加值 $\sum_{i=0}^{k} e(i)$,容易产生较大的累积误差。而增量式中由于消除了积分项,不需要做累加,控制量增量的确定仅与最近几次误差采样值有关,当存在计算误差或计算精度不足时,对控制量的计算影响较小。

(2) 增量式算法得出的是控制量的增量,系统工作会更安全。一旦计算机出现故障,会使控制信号 $\Delta u(k)$ 为零,此时执行机构的位置仍能保持前一步的位置 $u(k-1)$,因此对系统安全不会有大的影响。但是增量式算法需要有外部记忆的执行机构,比如步进电机。

(3) 控制系统在实现手动到自动无扰动切换时,在切换瞬时,必须首先将计算机的输出值设置为原始阀门开度 u_0,才能保证无冲击切换。由于增量式算式中不出现 u_0 项,则易于实现手动到自动的无冲击切换。

在实际应用中不可一概而论,应根据被控对象的实际情况进行选择。一般认为,当控制系统要求有较高的精度,或者是采用调节阀、伺服电动机时,应采用位置式算法来表征执行机构的具体位置;而执行机构如果是步进电机时应采用增量式算法来表征,控制器输出的控制量是相对于上次增加的控制量。

5.3.3 数字 PID 控制器算法的改进

用计算机来实现 PID 控制,不能只是单纯地把模拟 PID 控制离散化,这样并不能获得理想的控制效果,而是必须发挥计算机运算速度快、逻辑判断功能强、编程灵活等优势,从而使 PID 控制更加灵活多样,才能在控制性能上超过模拟控制器,能够更好地满足控制系统提出的各种要求。下面主要介绍几种改进的数字 PID 控制算法。

1. 积分分离 PID 控制算法

PID 控制器中积分的主要作用是消除静差,提高精度,但当扰动较大或给定值大幅度变化时,短时间内系统的输出会出现很大的偏差,致使积分部分幅值快速上升,再加上系统本身的惯性和滞后,这就会引起系统输出出现较大的超调和长时间的波动。特别对于温度、成分等变化缓慢的过程,这一现象更为严重,有可能引起系统振荡。为了防止这种现象发生,可以采用积分分离的方法来克服它。

积分分离 PID 控制算法的思想是当系统偏差较大时,取消积分作用,而当被调量接近给定值时,再加入积分作用,以减小静差,即

当 $|e(k)| \leqslant \beta$ 时使用 PID 数字控制器;当 $|e(k)| > \beta$ 时使用 PD 数字控制器。

积分分离 PID 的偏差阈值 β 作为控制算法中的一个可调参数，应根据具体对象的特性以及控制要求来决定，既不能过大，也不能过小。若 β 过大，则达不到积分分离的目的，如图 5.8 中的曲线 a；若 β 过小，系统偏差就有可能无法进入积分区，始终不能启用积分作用，系统只能进行 PD 控制，因此系统将会出现静差，如图 5.8 中的曲线 b；只有 β 的值适中，才能达到兼顾稳态偏差与动态品质的积分分离的目的，如图 5.8 中的曲线 c。积分分离 PID 控制算法的控制效果示意图如图 5.8 所示。

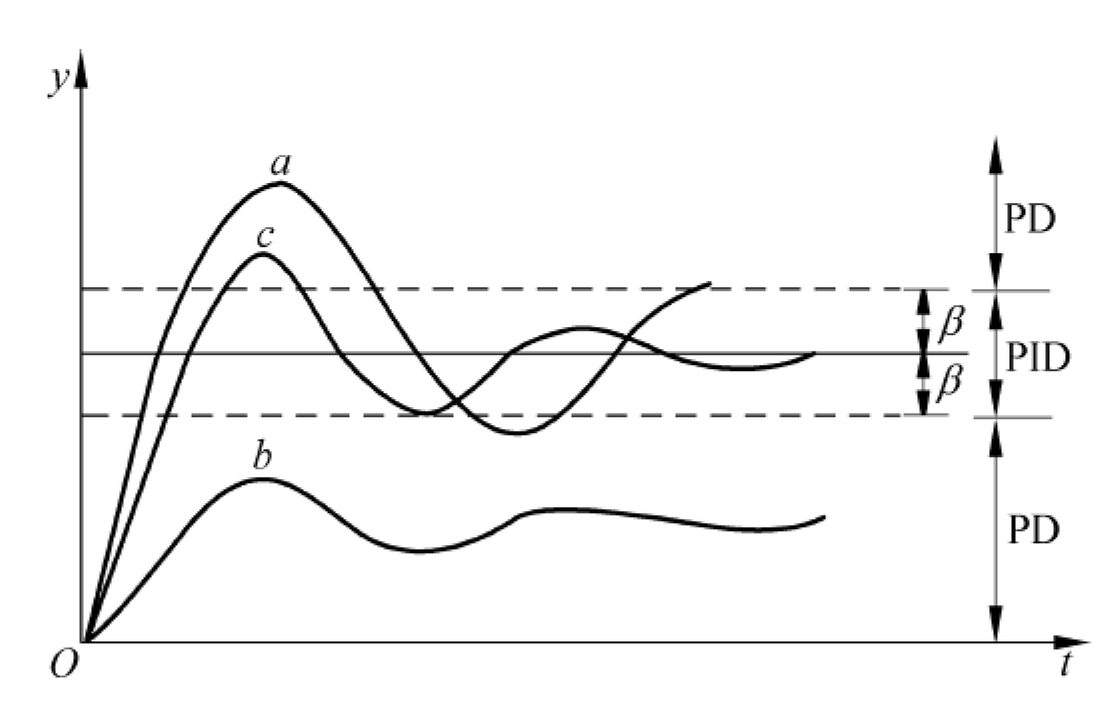


图 5.8 积分分离 PID 控制效果图

2. 不完全微分 PID 控制算法

微分作用有助于控制系统减小超调，克服振荡，使系统趋于稳定，同时还能加快系统动作，减小调整时间，有利于改善系统的动态性能。但是，对于具有高频扰动的生产过程，由于标准 PID 控制算式中的微分作用过于灵敏，容易引起控制过程振荡，降低了调节品质。另外对每个控制回路计算机的控制输出是短暂的，而执行机构的动作需要一定的时间来完成，如果输出值较大，执行机构在短暂的时间内一下子达不到应有的相应开度，输出将会失真。

为了克服这一缺点，同时又要使微分起作用，在标准 PID 控制算法中加入一个低通滤波器来抑制高频干扰，通常是一阶惯性环节，可以在整个 PID 控制器之后串联一个低通滤波器，也可以在微分部分串联一个低通滤波器，这就组成了不完全微分 PID 控制算法，如图 5.9 所示。

$E(s)$ → 标准PID调节器 → $U'(s)$ → $G_f(s)$ → $U(s)$

图 5.9 不完全微分 PID 控制器方框图

图中，低通滤波器 $G_f(s)$ 为一阶惯性环节，其传递函数为

$$D_f(s) = \frac{1}{T_f s + 1} \tag{5.60}$$

由传递函数定义可知

$$D_f(s) = \frac{U(s)}{U'(s)} = \frac{1}{T_f s + 1} \tag{5.61}$$

整理则有

$$U(s) + T_f s U(s) = U'(s) \tag{5.62}$$

对式(5.62)两边同时进行拉普拉斯反变换得

$$T_f\frac{du(t)}{dt}+u(t)=u'(t) \tag{5.63}$$

用后向差分法近似微分项，则可以求出式(5.63)的差分方程为

$$T_f\frac{u(k)-u(k-1)}{T}+u(k)=u'(k) \tag{5.64}$$

整理式(5.64)可得

$$u(k)=\frac{T_f}{T_f+T}u(k-1)+\frac{T}{T_f+T}u'(k)=\alpha u(k-1)+(1-\alpha)u'(k) \tag{5.65}$$

式中，$\alpha=\frac{T_f}{T_f+T}$；$u'(k)=K_p\left[e(k)+\frac{T}{T_i}\sum_{i=0}^{k}e(i)+T_d\frac{e(k)-e(k-1)}{T}\right]$（$u'(k)$为标准 PID 位置算式的输出）。

同理与标准 PID 控制算式一样，不完全微分 PID 也有增量式算法，即

$$\Delta u(k)=\alpha\Delta u(k-1)+(1-\alpha)\Delta u'(k) \tag{5.66}$$

式中，$\alpha=\frac{T_f}{T_f+T}$；$\Delta u'(k)=K_P[e(k)-e(k-1)]+K_Ie(k)+K_D[e(k)-2e(k-1)+e(k-2)]$（$K_I=K_PT/T_i$ 为积分系数，$K_D=K_PT_d/T$ 为微分系数）。

当偏差为阶跃变化时，可设数字微分调节器的输入为阶跃序列 $e(k)=1$，根据式(5.55)标准 PID 控制算式中微分的作用为 $u_d(k)=T_d\frac{e(k)-e(k-1)}{T}$，所以可知 $u_d(0)=\frac{T_d}{T}$，$u_d(1)=u_d(2)=\cdots=0$，也即标准 PID 中的微分只在第一个采样周期内起作用。而当使用不完全微分时，如果是将低通滤波器直接加在微分环节上，其不完全微分数字 PID 控制器的微分作用为 $u_d(s)=T_ds\frac{E(s)}{T_fs+1}$，其相应的微分方程为 $u_d(t)+T_f\frac{du_d(t)}{dt}=T_d\frac{de(t)}{dt}$，对其进行离散化，可得

$u_d(k)=\frac{T_f}{T_f+T}u_d(k-1)+\frac{T_d}{T_f+T}[e(k)-e(k-1)]$，由于偏差是阶跃变化，所以当 $e(k)=1$ 为单位阶跃函数时，则有 $u_d(0)=\frac{T_d}{T_f+T}e(0)=\frac{T_d}{T_f+T}$

$$u_d(1)=\frac{T_f}{T_f+T}u_d(0)+\frac{T_d}{T_f+T}[e(1)-e(0)]=\frac{T_fT_d}{(T_f+T)^2}$$

$$u_d(2)=\frac{T_f}{T_f+T}u_d(1)+\frac{T_d}{T_f+T}[e(2)-e(1)]=\frac{T_f^{\ 2}T_d}{(T_f+T)^3}$$

$$\vdots$$

显然

$$u_d(k)\neq 0,\quad k=1,2,\cdots,\quad 且\quad u_d(0)=\frac{T_d}{T_f+T}\leqslant\frac{T_d}{T}$$

根据上述计算推导过程，可做出标准 PID 控制算式和不完全微分 PID 控制算式在单位阶跃输入时输出的控制作用，如图 5.10 所示。

两种 PID 控制作用从图中比较可以看出，比例、积分部分输出是完全一样的，微分部分的输出差别较大。在输入误差 $e(k)$ 发生阶跃突变时，标准 PID 控制器中的微分作用只在第一个采样周期内有输出，并且微分作用较强，不能按照偏差的变化趋势在整个控制过程中起

作用,相比较而言,容易引起振荡以及微分饱和,系统产生溢出现象;而不完全微分作用则按指数规律逐渐衰减到零,可以延续几个采样周期,在较长的时间内仍有微分作用,并且第一个采样周期内的微分作用也比较弱,这样不易引起振荡,能获得较好的控制效果。

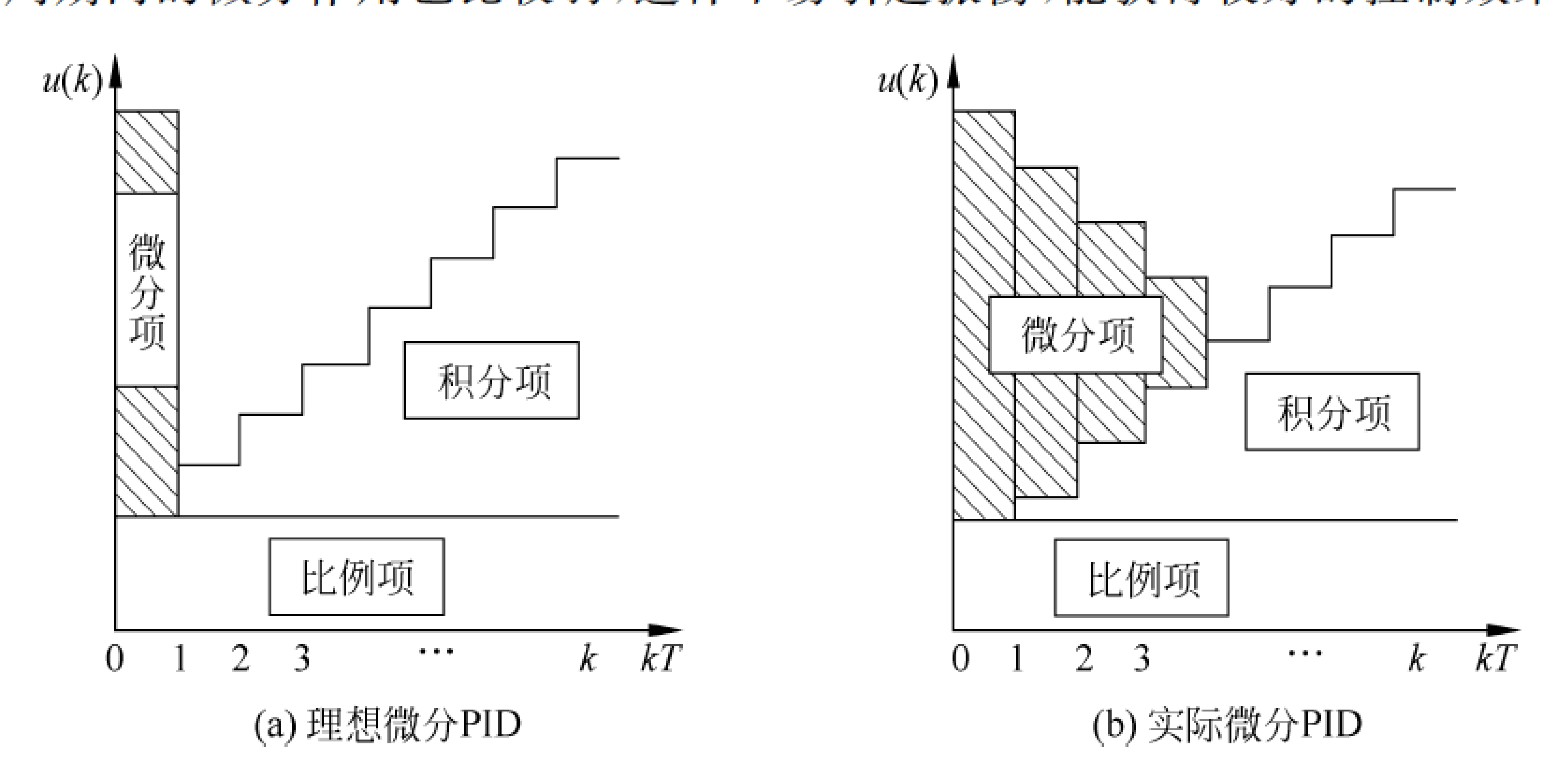


图 5.10 两种微分的 PID 控制作用阶跃响应比较

通过比较可知:

(1) 标准 PID 的控制品质比较差,因为微分作用仅仅局限于第一个采样周期,并且有一个大幅度的输出。一般工业用的执行机构,无法在较短的采样周期内跟踪较大的微分作用输出,而且理想微分很容易引进高频干扰。

(2) 不完全微分 PID 的控制品质较好,因为微分作用能缓慢地持续多个采样周期,使得一般的工业用执行机构能够较好地跟踪微分作用的输出。并且由于不完全微分 PID 算式中含有一个低通滤波器,具有滤波的能力,因此系统的抗干扰能力也比较强。

3. 微分先行 PID 控制算法

当控制系统的给定值发生阶跃变化时,微分动作将使控制量大幅度变化,这样不利于生产的稳定操作。在给定值频繁升降的场合,为了避免微分运算引起系统振荡,比如超调量过大,调节阀动作剧烈,可采用微分先行 PID 控制算法如图 5.11 所示。图中,γ 为微分增益系数。

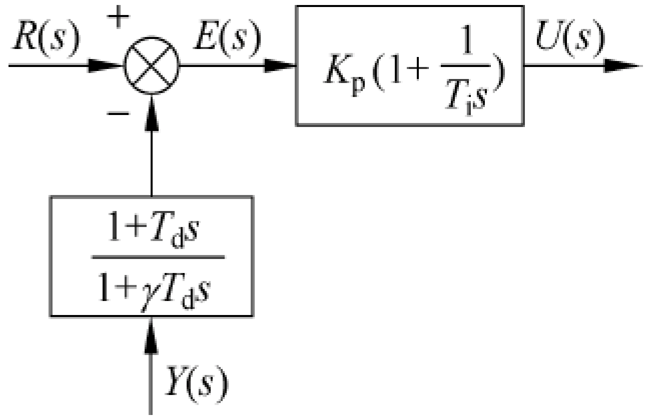


图 5.11 微分先行 PID 控制算法示意图

微分先行 PID 控制与普通 PID 控制的不同之处是微分先行 PID 只对系统的反馈量进行微分,而不对偏差微分,即对给定值无微分作用,而前面介绍的不完全微分 PID 中的微分作用,是对系统输入和反馈都有微分作用,属于偏差微分。对于串级控制的副回路而言,由于副控制回路的给定值是由主控制器给定的,也应该对其作微分处理,因此微分先行 PID 这种方法不适用,仍应按照偏差微分算式对偏差进行微分。

4. 带死区的 PID 控制算法

某些生产过程对控制精度要求不是很高,但控制过程要求尽量平稳,执行机构不要频繁动作。针对这一类系统,人们提出了一种带死区的 PID 控制算法。所谓带死区的 PID 控制算法,就是在计算机中人为地设置一个不灵敏区 ε,也即将输入的偏差信号设置一个适当范

围的死区。当偏差的绝对值 $|r(k)-y(k)|=|e(k)|\leqslant\varepsilon$ 时，不产生新的控制增量，控制量保持不变；当偏差的绝对值 $|r(k)-y(k)|=|e(k)|>\varepsilon$ 时，则进行正常的 PID 运算后输出。带死区的 PID 控制算法为

$$\Delta u(k)=\begin{cases}K_{\mathrm{P}}[e(k)-e(k-1)]+K_{\mathrm{I}}e(k)+K_{\mathrm{D}}[e(k)-2e(k-1)+e(k-2)] & |e(k)|>\varepsilon\\ 0 & |e(k)|\leqslant\varepsilon\end{cases}\tag{5.67}$$

死区 ε 是一个人为设置的可调参数，其具体数值一般应根据控制系统对被控量的控制精度的要求以及现场的试验结果来确定。如果 ε 太小，使调节动作过于频繁，达不到稳定被调对象的目的；如果 ε 太大，则系统将产生很大的滞后；当 $\varepsilon=0$ 时，即变成普通的 PID 控制。这种控制方式适用于控制精度要求不太高，控制动作尽可能少的场合。其控制系统方框图如图 5.12 所示，其中

$$P(k)=\begin{cases}e(k) & |r(k)-y(k)|=|e(k)|>\varepsilon\\ 0 & |r(k)-y(k)|=|e(k)|\leqslant\varepsilon\end{cases}\tag{5.68}$$

该系统实际上是一个非线性系统，死区是一个非线性环节，不能像线性环节一样随便地移到 PID 控制器的后面，对控制量输出设定一个死区，这样做的效果是完全不同的。

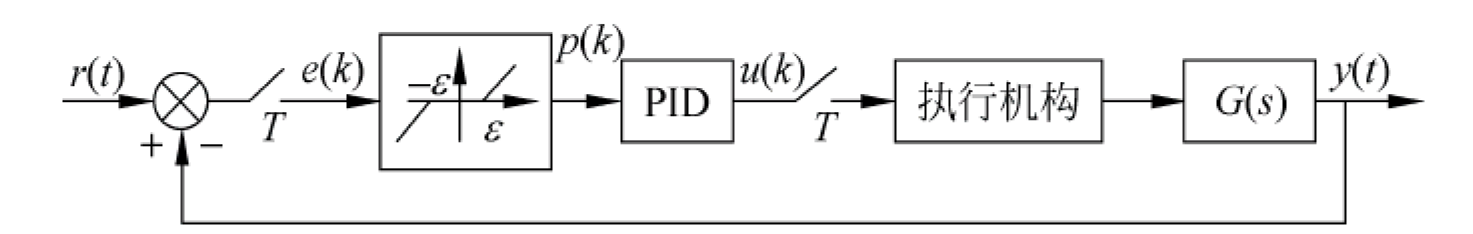


图 5.12　带死区的 PID 控制系框图

5. 提高积分项积分的精度

在 PID 控制算法中，积分项的作用是消除残差。为了减小残差，应提高积分项的运算精度，可将前面数字 PID 算式中的积分项替换为

$$\int_0^t e(t)\mathrm{d}t\approx T\sum_{i=0}^{k}\frac{e(i)+e(i-1)}{2}\tag{5.69}$$

即将原来的矩形积分改为梯形积分来近似计算。

5.3.4　数字 PID 控制器参数的整定

PID 控制器的设计一般分为两个步骤，首先是确定 PID 控制器的结构，在保证闭环控制系统稳定的前提下，尽可能的消除稳态误差。当确定好 PID 控制器的结构和控制规律以后，就要选择 PID 控制器的参数。系统控制质量的好坏主要取决于参数选取是否合理，设计确定 PID 参数的工作也称为 PID 控制器的参数整定。整定的实质是通过调整控制器的参数使其特性与被控对象的特性相匹配，达到要求的控制目标，以获得满意的控制效果。

PID 控制器参数整定的方法有很多，但都可以归纳为理论计算整定法和工程整定法两种。理论计算法以被控对象的数学模型为基础，要求知道各个环节的传递函数，计算比较复杂，并且所得结果的可靠性依赖于模型的精确程度，实际系统很难满足要求，所以这种方法

主要用于理论分析,在工程上用得并不多。实际中应用最多的是工程整定法,这是一种近似的基于经验和实验的方法,简单易行,便于实现,特别是不必依赖控制对象的数学模型,能解决控制工程中实际问题,是工程实际中经常采用的方法。比如后面介绍的扩充临界比例度法、扩充响应曲线法等工程整定法都在实际中得到了广泛的应用。

模拟PID控制器参数的整定是按照工艺对控制性能的要求,决定控制器的参数 K_p、T_i、T_d;而数字PID控制器与模拟控制器相比,除了需要整定PID参数外,还要确定系统的另一个重要参数,即系统的采样周期 T。

1. 采样周期 T 的确定

合理地确定采样周期 T 是数字控制系统设计的关键问题之一。由香农(Shannon)采样定理可知,只有当采样频率 $f_s \geqslant 2f_{max}$ 时,也即采样频率达到系统信号最高频率的两倍或两倍以上时,系统可由离散的采样信号真实地复现原来的信号。从而可知 $T \leqslant \pi/\omega_{max}$,这给出了采样周期的理论上限。

从理论上来讲,希望采样频率越高,也即采样周期越短,这样系统更接近于连续系统,使控制效果更好些,失真也越小。但从执行元件的特性要求来看,由于过程控制中通常采用电动或气动调节阀,它们的响应速度较低,如果采样周期过短,执行机构来不及响应,仍然达不到控制的目的,也就是要求在这段时间内,计算机的输出值不应变化,因此采样周期又要求必须大于这一时间。同时,还要考虑到计算机硬件和软件的制约,采样周期不能太小,如果采样周期太小,会增加计算机的负担,不利于发挥计算机的功能。当然,目前随着计算机技术的飞速发展,这方面的制约越来越小。

实际上在控制系统中,选择采样周期时,必须从以下几个方面综合考虑。

(1) 从被控对象的特性来看,采样周期应比对象的时间常数小得多,否则采样信息无法反应瞬变过程,并且快速系统的采样周期应取小,反之取大些。

(2) 执行机构的类型速度。当执行机构动作惯性大时,采样周期应取大些。否则,执行机构来不及反应控制器输出值的变化。并且执行机构的响应速度比较慢,那么过小的采样周期将失去意义。

(3) 控制算法的类型。不同的控制算法对采样周期有不同的影响,若控制器的算法比较复杂,则应选择较大的采样周期,以保证计算时间。

(4) 从控制系统的随动性和抗干扰性要求来看,希望采样周期小些,这样给定值的改变可以迅速地通过采样得到反映,而不致在随动控制中产生大的延时,并且干扰信号频率越高,采样频率最好也越高,以便实现快速抑制干扰。

(5) 控制回路数。从计算机所承担的工作量和每个调节回路的计算成本来看,一般则要求采样周期大些。特别当计算机用于多回路控制时,必须使每个回路的调节算法都有足够的时间来完成。也即采样周期的下限为计算机执行控制程序和输入输出所耗费的时间。

总之,各方面的因素不同对采样周期的要求是不同的,有时甚至是互相矛盾的,故必须根据具体情况和主要要求合理选择,以上仅是一些基本的选取原则。采样周期选择的理论计算法比较复杂,特别是被控制系统各环节时间常数等参数难以确定时,所以工程上较少应用理论计算法。实际的采样周期一般先按表5.3的经验数据进行选用,然后在现场调试运

行时进行修正。

表 5.3　采样周期的经验数据表

被测参数	采样周期 T/s	离散控制系统
流量	1～5	优先选用 1～2s
压力	3～10	优先选用 6～8s
液位	6～8	优先选用 7s
温度	15～20	取纯滞后时间常数
成分	15～20	优先选用 18s

2. 扩充临界比例度法

扩充临界比例度法是模拟调节器中使用的临界比例度法(也称稳定边界法)的扩充,是一种闭环整定的实验方法。它是简易工程整定方法之一,按该方法整定数字 PID 参数的步骤如下。

(1) 选择一个足够短的采样周期 $T_{\min}$。所谓足够短,具体地说就是采样周期选择为被控对象的纯滞后时间的 1/10 以下。

(2) 用所选定的采样周期使系统工作,这时数字控制器去掉积分作用和微分作用,只保留比例作用,形成闭环。给定输入作阶跃变化,并逐渐增大比例系数 K_P,也即逐渐减小比例度 $\delta(K_P=1/\delta)$,直到系统产生持续等幅振荡。此时的比例度即为临界比例度 δ_k,振荡周期称为临界振荡周期 T_k。

(3) 选择控制度。所谓控制度,就是数字控制系统与对应的模拟控制系统所对应的过渡过程的误差平方的积分之比,即

$$\text{控制度}=\frac{\left[\int_0^{\infty} e^2(t)\,\mathrm{d}t\right]_{\text{数字控制}}}{\left[\int_0^{\infty} e^2(t)\,\mathrm{d}t\right]_{\text{模拟控制}}}$$

因为数字控制系统是断续控制,而模拟控制系统是连续控制,所以对于同一个系统,采用相同的控制规律,数字控制系统的品质总是低于模拟控制系统的品质,所以,控制度总是大于 1。控制度越大,表明相应的数字控制系统品质越差。

控制度是数字控制器和模拟控制器控制效果相比较的一种性能评价指标,实际应用中并不需要计算出两个误差的平方积分,控制度仅表示控制效果的物理概念。例如,当控制度为 1.05 时,数字控制器和模拟控制器的控制效果基本相同;当控制度为 2.0 时,数字控制比模拟控制的控制质量差一倍。控制器参数随控制度的不同而略有区别,为使数字控制器的控制效果尽可能的接近模拟控制器,应使控制度接近 1.05。

(4) 根据所选定的控制度,然后查表 5.4 即可求出采样周期 T、比例系数 K_p、积分时间常数 T_i 和微分时间常数 T_d。

(5) 根据所求得的参数,将系统投入运行,观察控制效果,如果效果性能欠佳,可适当加大比例度,重新求取各个参数,继续观察控制效果,直到得到满意的控制效果。

表 5.4 扩充临界比例度法参数整定计算表

控制度	控制规律	T/T_k	K_p/δ_k	T_i/T_k	T_d/T_k
1.05	PI	0.03	0.53	0.88	—
	PID	0.014	0.63	0.49	0.14
1.20	PI	0.05	0.49	0.91	—
	PID	0.043	0.47	0.47	0.16
1.50	PI	0.14	0.42	0.99	—
	PID	0.09	0.34	0.43	0.20
2.00	PI	0.22	0.36	1.05	—
	PID	0.16	0.27	0.40	0.22
连续时间控制器	PI	—	0.57	0.83	—
	PID	—	0.70	0.50	0.13

3. 扩充响应曲线法

在扩充比例度法中,不需要知道对象的动态特性,而是直接在闭环系统中进行整定的。与闭环整定方法不同,扩充响应法是一种开环整定方法。扩充响应曲线法是对模拟调节器中使用的响应曲线法的扩充,它是建立在已知系统的动态特性的基础上,也是一种实践经验法。其整定步骤如下。

(1) 首先是断开数字控制器,即不将数字控制器接入控制系统,让系统处于手动状态下,将被调量调节到给定值附近,并使之稳定下来。然后突然改变给定值,也即给对象输入一个阶跃信号。

(2) 用仪表记录被调量在阶跃输入下的整个变化过程曲线,如图 5.13 所示。

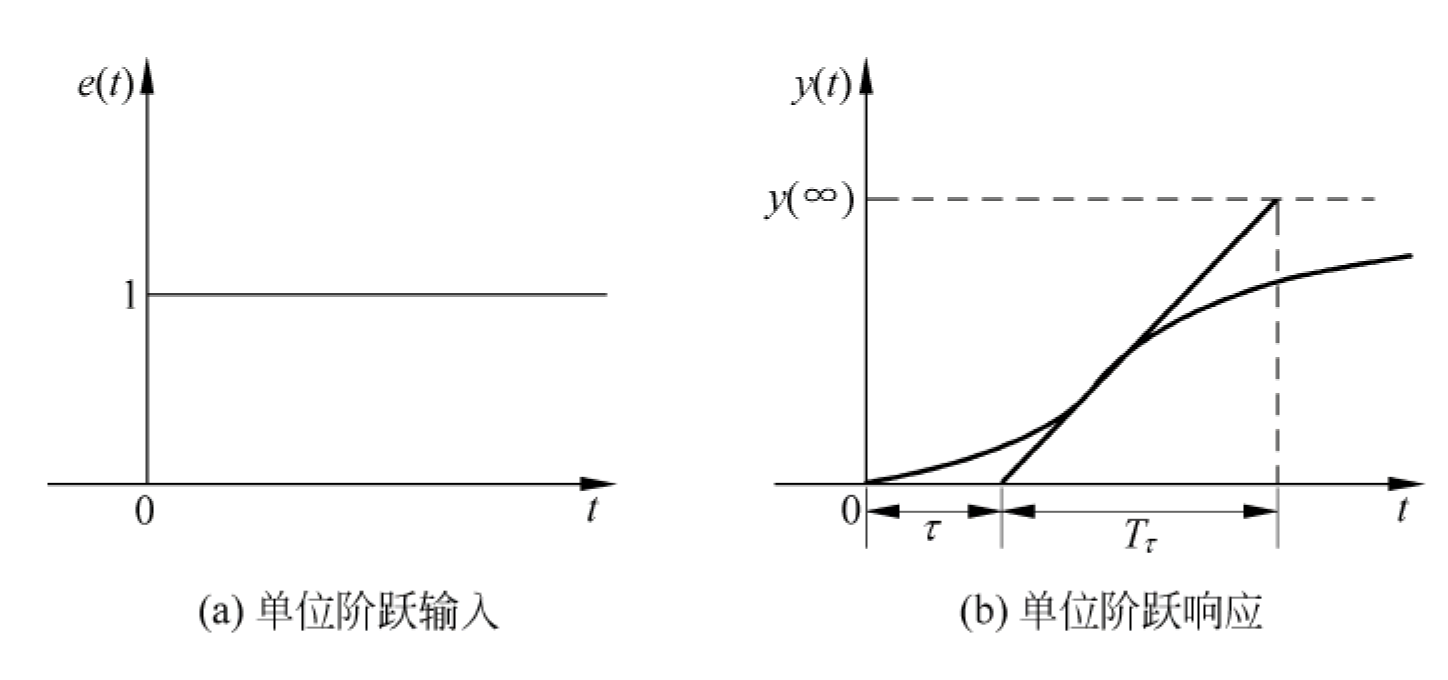


图 5.13 被控对象阶跃响应曲线

(3) 在曲线最大斜率处做切线,求得被控对象的时间常数 T_τ、滞后时间 τ 以及它们的比值 T_τ/τ。

(4) 选择某一控制度,查表 5.5 即可求得数字 PID 的整定参数 K_p、T_i、T_d 以及采样周期 T。

表 5.5 扩充响应曲线法参数整定计算表

控制度	控制规律	T	K_p	T_i	T_d
1.05	PI	0.1τ	$0.84T_\tau/\tau$	3.4τ	
	PID	0.05τ	$1.15T_\tau/\tau$	2.0τ	0.45τ
1.2	PI	0.2τ	$0.78T_\tau/\tau$	3.6τ	
	PID	0.16τ	$1.0T_\tau/\tau$	1.9τ	0.55τ
1.50	PI	0.5τ	$0.68T_\tau/\tau$	3.9τ	
	PID	0.34τ	$0.85T_\tau/\tau$	1.62τ	0.65τ
2.0	PI	0.8τ	$0.57T_\tau/\tau$	4.2τ	
	PID	0.6τ	$0.6T_\tau/\tau$	1.5τ	0.82τ

4. PID 归一参数整定法

控制器参数的整定是一项繁琐而又费时的工作，国内外学者在数字 PID 参数整定方面作了大量的工作。Robers P.D 在 1974 年提出了一种简化扩充临界比例度整定法，该方法是在扩充临界比例度法的基础上，进行简化而来的。由于该方法只需整定一个参数即可，故称其为归一参数整定法。

根据齐格勒-尼柯尔斯经验公式，令 $T=0.1T_k$，$T_i=0.5T_k$，$T_d=0.125T_k$，T_k 为纯比例作用下的临界振荡周期，又已知增量型 PID 控制的公式为

$$\Delta u(k) = K_P\left\{e(k)-e(k-1)+\frac{T}{T_i}e(k)+\frac{T_d}{T}[e(k)-2e(k-1)+e(k-2)]\right\}$$

代入可对其进行简化得

$$\Delta u(k) = K_P[2.45e(k)-3.5e(k-1)+1.25e(k-2)]$$

这样，要整定的四个参数就简化为只要整定一个参数 K_P，在线调整 K_P，观察控制效果，直到满意为止。该方法为实现简易的自整定控制带来了方便。

5. 试凑法

在 PID 参数整定方法中，最基本和最简单的方法为试凑法。试凑法是通过模拟或实际的系统闭环运行情况，观察系统的响应曲线，然后再根据各调节参数对系统响应的大致影响，从一组初始 PID 参数开始，反复试凑，不断修改参数，以达到满意的效果为止，从而确定 PID 控制器的参数，这也是目前实际工程应用中最为广泛的一种 PID 控制器参数整定方法。

试凑时要注意 PID 控制器各参数对系统影响的特点，在实践中总结出了如下的规律。

(1) 增大开环比例系数，一般将加快系统的影响速度，在有静差的情况下则有利于减小静差；但过大的比例系数又会加大系统超调，甚至产生振荡，使系统不稳定。

(2) 增大积分时间，一般有利于减小超调，减少振荡，使系统的稳定性增加，但是系统静差的消除将随之变慢。

(3) 增大微分时间也有利于加快系统的响应，使超调量减小，稳定性增加，但系统对干扰的抑制能力减弱，对扰动有敏感响应的系统不宜采用微分环节。

在试凑时，对参数的调整遵循的原则是先比例后积分再微分的整定步骤，其步骤如下

所述。

(1) 整定比例部分。先置 PID 控制器中的 $T_i=\infty$、$T_d=0$,使之成为比例控制器,再将比例系数由小变大,并观察相应的系统响应,使系统的过渡过程达到 4∶1 的衰减振荡和较小的静差,得到反应快、超调小的响应曲线。如果系统没有静差或静差已小到允许范围内,并且已达到 4∶1 衰减的响应曲线,那么只需用比例控制器即可,最优比例系数就由此确定。

(2) 加入积分环节。如果只用比例控制,系统的静差不能满足设计要求,则需加入积分环节。整定时,先将第一步整定得到的比例系数减小 10%～20%,以补偿因加入积分作用而引起的系统稳定性下降,然后由大到小调节 T_i,使系统在保持良好动态性能的情况下能够消除静差。这一步可以反复进行,以期得到满意的控制效果。

(3) 加入微分环节。如果在使用 PI 控制消除了静差,但动态过程仍不满意,比如调节时间过长或超调量过大,则可以加入微分环节,来构成 PID 控制器。在整定时,先置 T_d 为零,然后在第二步整定的基础上再逐步增大 T_d,同时相应地改变比例系数 K_p 和积分时间 T_i,逐步试凑以获得最佳的控制效果和控制参数。

5.4 数字控制器的离散化设计方法

计算机控制系统的连续化设计是以连续时间系统的控制理论为基础的,在计算机上进行数字模拟来实现的,对于采样周期远小于被控对象时间常数的生产过程,把离散时间系统近似为连续时间系统,用技术成熟的连续化设计技术并把它移植到计算机上来实现,可以达到满意的控制效果。但这是以采样周期足够小为前提的,这是许多实际系统难以满足的。当根据控制任务的要求,需要选择较大的采样周期或对控制质量有较高要求时,就不能采用连续化设计方法,必须从被控对象的特性出发,根据系统的性能要求,直接根据计算机控制理论设计出计算机控制系统的数字控制器,这种方法称为离散化设计法或者直接数字设计法。离散化设计法比连续化设计法更具有一般的意义,它绕过了模拟控制器离散化过程,也避免了采样周期对系统动态性能的影响,完全根据采样控制系统的特点进行分析和综合,并导出相应的控制规律,是目前较为广泛采用的计算机控制系统的设计方法,所用的数学工具是 Z 变换和 z 传递函数。后面介绍的最小拍控制系统的设计就是采用离散化设计方法的一个典型实例。

5.4.1 数字控制器离散化设计步骤

图 5.14 是典型的计算机控制系统结构图,其中 $G_c(s)$ 为被控对象的传递函数,$H(s)=\dfrac{1-\mathrm{e}^{-Ts}}{s}$ 为零阶保持器,$G(z)=Z[H(s)G_c(s)]=Z\left[\dfrac{1-\mathrm{e}^{-Ts}}{s}G_c(s)\right]$ 定为广义对象的脉冲传递函数,$\Phi(z)$ 为系统的闭环脉冲传递函数,$D(z)$ 表示是需要设计的数字控制器,T 为采样周期。

由图 5.14 可知,该系统的闭环脉冲传递函数为

$$\Phi(z)=\frac{Y(z)}{R(z)}=\frac{D(z)G(z)}{1+D(z)G(z)} \tag{5.70}$$

其误差脉冲传递函数为

$$\Phi_e(z)=\frac{E(z)}{R(z)}=\frac{R(z)-Y(z)}{R(z)}=1-\Phi(z)=\frac{1}{1+D(z)G(z)} \tag{5.71}$$

由式(5.70)可得数字控制器的脉冲传递函数为

$$D(z)=\frac{1}{G(z)}\frac{\Phi(z)}{1-\Phi(z)}=\frac{\Phi(z)}{G(z)\Phi_e(z)} \tag{5.72}$$

式中，$G(z)$为系统的固有特性无法改变，而离散化设计的目标就是根据预期的控制指标，直接设计出满足要求的数字控制器 $D(z)$，因此只要确定满足系统性能指标的闭环脉冲传递函数 $\Phi(z)$即可。由此可得出数字控制器的离散化设计步骤如下。

(1) 当设计要求一旦确定，根据控制系统的性能指标要求及其他约束条件，确定所需的闭环脉冲传递函数 $\Phi(z)$。

(2) 根据被控对象和零阶保持器的传递函数求出广义对象的脉冲传递函数 $G(z)$。

(3) 根据式(5.72)可推出数字控制器的脉冲传递函数 $D(z)$。

(4) 根据 $D(z)$求取控制算法的递推计算公式。

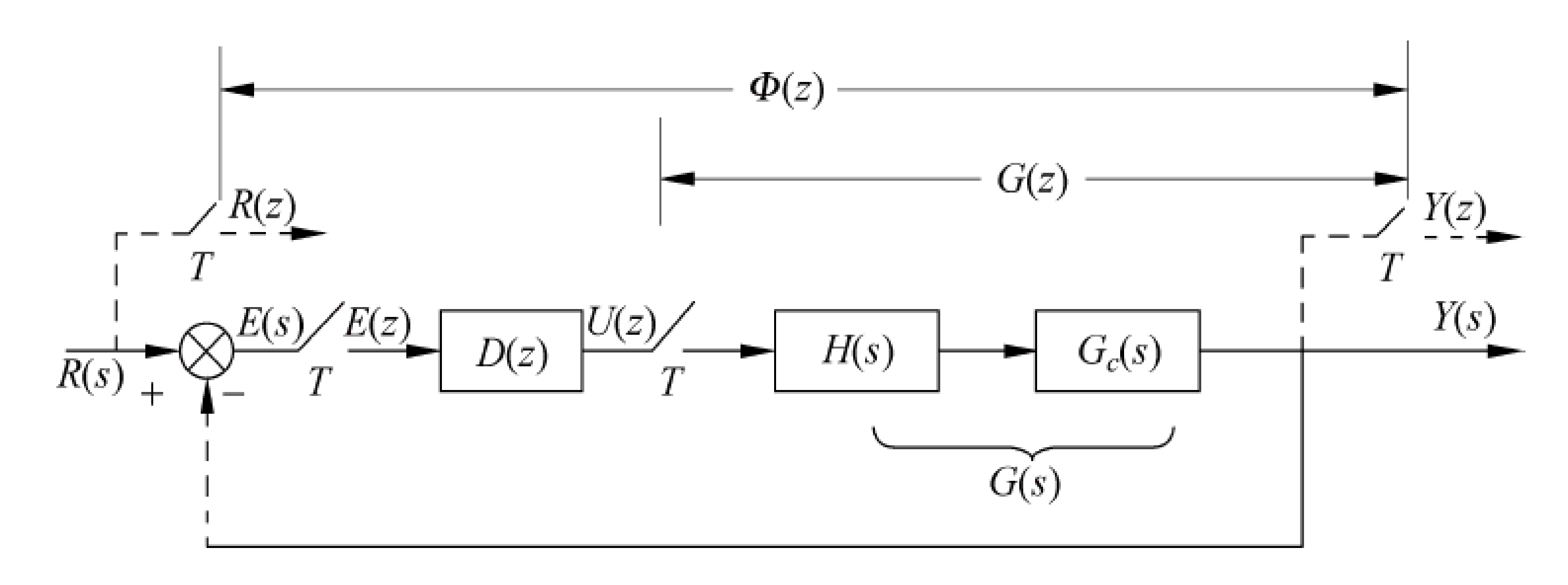


图 5.14　典型计算机控制系统结构图

由 $D(z)$求取控制算法的递推公式可按以下方法来实现。

设数字控制器 $D(z)$的一般形式为

$$D(z)=\frac{U(z)}{E(z)}=\frac{\sum_{i=0}^{m}b_i z^{-i}}{1+\sum_{i=1}^{n}a_i z^{-i}},\quad m\leqslant n \tag{5.73}$$

由式(5.73)可知数字控制器的输出 $U(z)$为

$$U(z)=\sum_{i=0}^{m}b_i z^{-i}E(z)-\sum_{i=1}^{n}a_i z^{-i}U(z) \tag{5.74}$$

因此，数字控制器 $D(z)$的计算机控制算法为

$$u(k)=\sum_{i=0}^{m}b_i e(k-i)-\sum_{i=1}^{n}a_i u(k-i) \tag{5.75}$$

按照式(5.75)就可编写数字控制器的控制算法程序。

5.4.2　最少拍控制系统的设计

在自动调节系统中，当偏差存在时，人们总是希望尽快地消除偏差，使输出跟随输入变化，或者在有限的几个采样周期内即可达到平衡。特别数字随动系统中，要求系统的输出值

尽快地跟踪给定值的变化,最少拍控制就是为满足这一要求的一种离散化数字设计方法。

计算机控制系统常以多个采样周期来计时,一个采样周期称为一拍。最少拍控制实际上是一种时间最优控制,通常指系统在典型输入信号(比如阶跃信号、速度信号、加速度信号等)作用下,经过最少个采样周期使系统输出的稳态误差为零。因此最少拍控制有时也称为最少拍无差系统或者最少拍随动系统。

最少拍控制系统的性能指标主要有以下几项。

(1) 闭环系统必须是稳定的,即稳定性要求。

(2) 对于特定的参考输入,在达到稳态后,系统在采样点处无静差,也即采样点的输出值精准地跟踪输入信号,即准确性要求。这有两种情况:一种是要求在采样点上的稳态误差等于零,采样点之间的稳态误差不为零,这就是常说的最少拍有纹波系统;另一种是不仅在采样点上,而且在采样点之间的稳态误差都等于零,这通常称为最少拍无纹波系统。

(3) 在各种使系统在有限拍内达到稳态的设计中,调节时间应最短,即系统准确跟踪输入信号所需要的采样周期数为最少,即快速性要求。

(4) 设计出来的数字控制器必须是物理上可实现的,即可实现性要求。

1. 闭环脉冲传递函数 $\boldsymbol{\Phi}(z)$ 的确定

根据最少拍系统的要求,也即在典型的控制输入信号作用下,系统能在最少几个采样周期内达到无静差的稳态,也即闭环脉冲传递函数 $\Phi(z)$ 具有以下形式

$$\Phi(z)=m_1 z^{-1}+m_2 z^{-2}+m_3 z^{-3}+\cdots+m_n z^{-n} \tag{5.76}$$

式中,n 是可能情况下的最小正整数。这一传递函数形式也表明闭环系统的脉冲响应在 n 个采样周期后变为零,即系统在 n 拍后到达稳态。

由图 5.14 可知系统的误差脉冲传递函数 $\Phi_e(z)$ 为

$$\Phi_e(z)=\frac{E(z)}{R(z)}=\frac{R(z)-Y(z)}{R(z)}=1-\Phi(z)=\frac{1}{1+D(z)G(z)} \tag{5.77}$$

所以系统输出的偏差为

$$E(z)=R(z)\Phi_e(z)=[1-\Phi(z)]R(z) \tag{5.78}$$

一般的控制系统常见的典型输入有以下三种。

(1) 单位阶跃输入: $r(t)=1(t)$, $R(z)=\dfrac{z}{z-1}=\dfrac{1}{1-z^{-1}}$

(2) 单位速度输入: $r(t)=t$, $R(z)=\dfrac{Tz^{-1}}{(1-z^{-1})^2}$

(3) 单位加速度输入: $r(t)=\dfrac{1}{2}t^2$, $R(z)=\dfrac{\frac{1}{2}T^2z^{-1}(1+z^{-1})}{(1-z^{-1})^3}$

其中,T 为采样周期。可以看出它们 Z 变换的一般形式可以表示为

$$R(z)=\frac{A(z)}{(1-z^{-1})^q} \tag{5.79}$$

式中,q 为正整数,是典型输入信号的阶次,对于不同的输入,只是 q 不同而已,一般只讨论 $q=1$、2、3 的情况;$A(z)$是不包括$(1-z^{-1})$因式的关于 z^{-1} 的多项式。

将式(5.79)代入式(5.78)可得

$$E(z)=\Phi_e(z)\frac{A(z)}{(1-z^{-1})^q} \tag{5.80}$$

根据 z 变换的终值定理可求出系统的稳态误差为

$$\begin{aligned}e(\infty)&=\lim_{z\to1}(1-z^{-1})E(z)\\&=\lim_{z\to1}(1-z^{-1})R(z)\Phi_e(z)\\&=\lim_{z\to1}(1-z^{-1})\frac{A(z)}{(1-z^{-1})^q}\Phi_e(z)\end{aligned} \tag{5.81}$$

由于 $A(z)$不包括$(1-z^{-1})$的因子，从准确性的要求来看，为了使系统的稳态误差为零，因此 $\Phi_e(z)$必须含有因子$(1-z^{-1})^q$，则 $\Phi_e(z)$可表示为下列形式

$$\Phi_e(z)=(1-z^{-1})^pF(z) \tag{5.82}$$

式中，$p\geqslant q$，q 是典型输入函数 $R(z)$中分母$(1-z^{-1})$因子的阶次，$F(z)$是不含$(1-z^{-1})$因子的关于 z^{-1}的待定系数的有限多项式。选择合适的 $\Phi_e(z)$就是选择合适的 p 值以及 $F(z)$。

将偏差 $E(z)$的 z 变化展开成如下形式

$$E(z)=\sum_{k=0}^{+\infty}e(kT)z^{-k}=e(0)+e(T)z^{-1}+e(2T)z^{-2}+\cdots+e(kT)z^{-k}+\cdots \tag{5.83}$$

从式(5.83)可知，根据快速性要求，要使系统的偏差尽快地为零，应使式(5.83)中关于 z^{-1} 的多项式数为最少，因此式(5.82)中的 p 应选择为

$$p=q \tag{5.84}$$

若要使 $E(z)$以及设计的数字控制器最简单、阶数最低，必须取 $F(z)=1$，这就是说，使 $F(z)$中不含 z^{-1}的因子，$\Phi_e(z)$才能使 $E(z)$中关于 z^{-1}的项数最少，调节时间最短。这样，$E(z)$的阶次仅仅取决于 $R(z)$的分子多项式 $A(z)$。因此由式(5.82)可得 $\Phi_e(z)$为

$$\Phi_e(z)=(1-z^{-1})^q \tag{5.85}$$

所以

$$\Phi(z)=1-\Phi_e(z)=1-(1-z^{-1})^q \tag{5.86}$$

又由于 $G(z)$已知，所以由式(5.72)可求得数字控制器 $D(z)$为

$$D(z)=\frac{1}{G(z)}\frac{\Phi(z)}{1-\Phi(z)}=\frac{1-(1-z^{-1})^q}{G(z)(1-z^{-1})^q} \tag{5.87}$$

2. 典型输入下最少拍系统的设计

1）单位阶跃输入

单位阶跃输入函数为 $r(t)=1(t)$，其 Z 变换为 $R(z)=\frac{z}{z-1}=\frac{1}{1-z^{-1}}$，即 $q=1$，由式(5.85)、式(5.86)可知

$$\Phi_e(z)=1-z^{-1},\quad \Phi(z)=1-\Phi_e(z)=z^{-1}$$

则由式(5.78)可求出偏差的 Z 变换为

$$E(z)=R(z)\Phi_e(z)=1$$

结合式(5.83)展开有

$$E(z)=1=1\cdot z^0+0\cdot z^{-1}+0\cdot z^{-2}+\cdots+0\cdot z^{-k}+\cdots$$

即 $e(0)=1$，$e(1)=e(2)=\cdots=0$，这说明系统只需一拍，也即一个采样周期，输出就能准确地跟踪输入，使系统的偏差为零，过渡过程结束。

由闭环脉冲传递函数 $\Phi(z)$ 可求得输出 $Y(z)$ 并用长除法展开为

$$Y(z)=R(z)\Phi(z)=\frac{1}{1-z^{-1}}z^{-1}=z^{-1}+z^{-2}+z^{-3}+\cdots$$

这也说明 $y(0)=0, y(1)=y(2)=\cdots=1$，即经过一个采样周期输出和输入完全跟踪一致，调整时间为一拍。

2）单位速度输入

单位速度输入函数为 $r(t)=t$，其 Z 变换为 $R(z)=\dfrac{Tz^{-1}}{(1-z^{-1})^2}$，即 $q=2$

由式(5.85)、式(5.86)可知

$$\Phi_e(z)=(1-z^{-1})^2,\quad \Phi(z)=1-\Phi_e(z)=2z^{-1}-z^{-2}$$

则由式(5.78)可求出偏差的 Z 变换为

$$E(z)=R(z)\Phi_e(z)=Tz^{-1}$$

结合式(5.83)展开有

$$E(z)=1=0\cdot z^0+T\cdot z^{-1}+0\cdot z^{-2}+\cdots+0\cdot z^{-k}+\cdots$$

即 $e(0)=0, e(1)=T, e(2)=e(3)=\cdots=0$，这说明系统只需经过两拍，也即两个采样周期，输出就可以无差地跟踪输入的变化，使系统的偏差为零，过渡过程结束。

由闭环脉冲传递函数 $\Phi(z)$ 可求得输出 $Y(z)$ 并用长除法展开为

$$Y(z)=R(z)\Phi(z)=\frac{Tz^{-1}(2z^{-1}-z^{-2})}{(1-z^{-1})^2}=2Tz^{-2}+3Tz^{-3}+4Tz^{-4}+\cdots$$

这也说明 $y(0)=0, y(1)=0, y(2)=2T, y(3)=3T, y(4)=4T, \cdots$，即经过两个采样周期输出和输入完全跟踪一致，系统达到稳态，其调节时间为 $2T$。

3）单位加速度输入

单位加速度输入函数为 $r(t)=\dfrac{1}{2}t^2$，其 z 变换为 $R(z)=\dfrac{\frac{1}{2}T^2z^{-1}(1+z^{-1})}{(1-z^{-1})^3}$ 即 $q=3$

由式(5.85)、式(5.86)可知

$$\Phi_e(z)=(1-z^{-1})^3,\quad \Phi(z)=1-\Phi_e(z)=3z^{-1}-3z^{-2}+z^{-3}$$

则由式(5.78)可求出偏差的 z 变换为

$$E(z)=R(z)\Phi_e(z)=\frac{1}{2}T^2z^{-1}+\frac{1}{2}T^2z^{-2}$$

结合式(5.83)展开有

$$E(z)=0\cdot z^0+\frac{T^2}{2}\cdot z^{-1}+\frac{T^2}{2}\cdot z^{-2}+0\cdot z^{-3}+0\cdot z^{-4}+\cdots$$

即 $e(0)=0, e(1)=\dfrac{T^2}{2}, e(2)=\dfrac{T^2}{2}, e(3)=e(4)=\cdots=0$，这说明系统只需三拍，也即三个采样周期，输出就可以无差地跟踪输入，达到稳态。此时输出 $Y(z)$ 并用长除法展开为

$$\begin{aligned}Y(z)=R(z)\Phi(z)&=\frac{\frac{T^2}{2}z^{-1}(1+z^{-1})(3z^{-1}-3z^{-2}+z^{-3})}{(1-z^{-1})^3}\\&=\frac{T^2}{2}(0\cdot z^0+0\cdot z^{-1}+3z^{-2}+9z^{-3}+16z^{-4}+\cdots)\end{aligned}$$

这也说明 $y(0)=0, y(1)=0, y(2)=\dfrac{3T^2}{2}, y(3)=\dfrac{9T^2}{2}, y(4)=\dfrac{16T^2}{2}, \cdots$，即经过三个采样周

期输出和输入完全跟踪一致，系统的调节时间为 $3T$。

综合以上可知三种典型输入下的最少拍控制系统的分析结论如表 5.6 所示。

表 5.6　典型输入下最少拍系统设计表

$r(t)$	$R(z)$	q	$\Phi_e(z)$	$\Phi(z)=1-\Phi_e(z)$	$D(z)=\dfrac{\Phi(z)}{\Phi_e(z)G(z)}$	$E(z)=\Phi_e(z)\cdot R(z)$	最少拍数
$1(t)$	$\dfrac{z}{z-1}$ $\dfrac{1}{(1-z^{-1})}$	1	$1-z^{-1}$	z^{-1}	$\dfrac{z^{-1}}{(1-z^{-1})G(z)}$	1	T
t	$\dfrac{Tz}{(z-1)^2}$ $\dfrac{Tz^{-1}}{(1-z^{-1})^2}$	2	$(1-z^{-1})^2$	$2z^{-1}-z^{-2}$	$\dfrac{2z^{-1}-z^{-2}}{(1-z^{-1})^2G(z)}$	Tz^{-1}	$2T$
$\dfrac{t^2}{2}$	$\dfrac{T^2z(z+1)}{2(z-1)^3}$ $\dfrac{T^2z^{-1}(1+z^{-1})}{2(1-z^{-1})^3}$	3	$(1-z^{-1})^3$	$3z^{-1}-3z^{-2}+z^{-3}$	$\dfrac{3z^{-1}-3z^{-2}+z^{-3}}{(1-z^{-1})^3G(z)}$	$\dfrac{T^2}{2}(z^{-1}+z^{-2})$	$3T$

3. 最少拍控制器的局限性

1）最少拍无差系统对典型输入的适应性差

由前面典型输入的最少拍系统的设计可以看出，最少拍控制器设计时 $\Phi(z)$ 或 $\Phi_e(z)$ 的选取与典型输入信号的形式密切相关，即对于不同的输入要求用不同的闭环脉冲传递函数，这样设计出的控制器对各种典型输入信号的适应性差。

例如，当 $\Phi(z)$ 是按单位速度输入函数设计时有

$$\Phi_e(z)=(1-z^{-1})^2,\quad \Phi(z)=1-\Phi_e(z)=2z^{-1}-z^{-2}$$

从前面的讨论可知，该系统的输出经过两拍就可以在采样点处无差地跟踪上输入的变化，输出响应如图 5.15(a) 所示。针对单位速度输入设计的这个最少拍系统，对其他输入是否还能成为最少拍吗？下面就对单位阶跃和单位加速度输入下的该系统的响应情况进行分析。

单位阶跃输入时

$$r(t)=1(t),\quad R(z)=\frac{z}{z-1}=\frac{1}{1-z^{-1}}$$

输出量为

$$Y(z)=R(z)\Phi(z)=\frac{2z^{-1}-z^{-2}}{1-z^{-1}}=2z^{-1}+z^{-2}+z^{-3}+z^{-4}+\cdots$$

也即输出序列为 $y(0)=0, y(1)=2, y(2)=y(3)=\cdots=1$，输出响应如图 5.15(b) 所示。从图中可以看出，针对单位速度设计的最少拍系统在单位阶跃的输入下，经过两个采样周期就可以稳定在设定值上，但是超调严重，在第一个采样点上有 100% 的超调量。

单位加速度输入时

$$r(t)=\frac{1}{2}t^2,\quad R(z)=\frac{\frac{1}{2}T^2z^{-1}(1+z^{-1})}{(1-z^{-1})^3}$$

输出量为

$$Y(z)=R(z)\Phi(z)=(2z^{-1}-z^{-2})\frac{z^{-1}(1+z^{-1})}{2(1-z^{-1})^3}$$
$$=T^2z^{-2}+3.5T^2z^{-3}+7T^2z^{-4}+11.5T^2z^{-5}+\cdots$$

也即输出序列为 $y(0)=0, y(1)=0, y(2)=T^2, y(3)=3.5T^2, y(4)=7T^2, y(5)=11.5T^2, \cdots$，此时单位加速度输入 $r(t)=\frac{1}{2}t^2$ 的采样函数为 $r(kT)=\frac{1}{2}(kT)^2$，即输入序列为 $r(0)=0$，$r(1)=0.5T^2, r(2)=2T^2, r(3)=4.5T^2, r(4)=8T^2, r(5)=12.5T^2, \cdots$，比较输入序列和输出序列可知从第 2 拍开始，输出与输入的误差为 T^2，输出响应如图 5.15(c)所示。从图中可以看出，针对单位速度设计的最少拍系统在单位加速度的输入下，始终存在有静差。

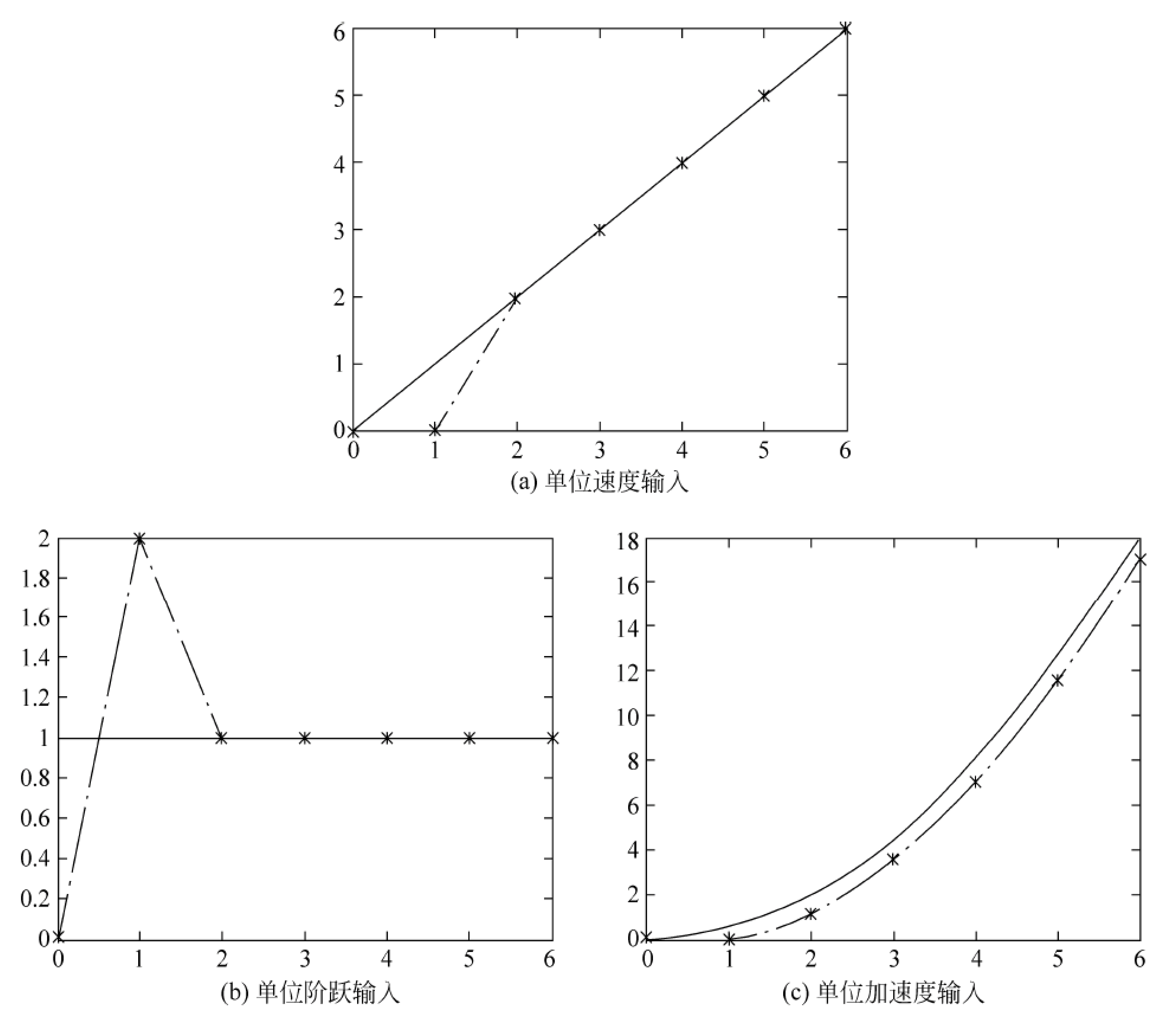


图 5.15 按单位速度输入设计的最少拍系统对不同输入的响应曲线图

由以上分析可以看出，按照某种典型输入设计的最少拍系统，当输入形式改变时，对其他函数的响应并不是最少拍系统，系统的性能变坏，输出响应也不一定理想，可见最少拍系统对输入函数的适应性较差。一般来说，针对某一种典型输入设计的最少拍系统，用于次数较低的输入函数时，系统将会出现较大的超调，响应时间也会增加，但是稳态时在采样时刻的误差为零。相反地，当用于次数较高的输入函数时，输出将不能完全跟踪输入，存在静差。由此可见，一种典型的最少拍闭环脉冲传递函数 $\Phi(z)$ 只适应一种特定的输入而不能适应于各种输入，也即最少拍控制系统对输入信号变化的适应性较差。

2）最少拍控制器物理上的可实现性

在前面讨论最少拍控制器 $D(z)$ 的设计方法时，并没有考虑最少拍控制器 $D(z)$ 的物理上是否能实现的问题。所谓物理的可实现性是指控制器当前的输出信号，应只与当前时刻的输入信号以及过去时刻的输入信号和输出信号有关，而与将来的输入信号无关。这是因为将来的信号是未知的，不能用于计算现在的控制量。这就要求数字控制器 $D(z)$ 中不能有 z 的正幂项。设数字控制器 $D(z)$ 的一般表达式为

$$D(z)=\frac{U(z)}{E(z)}=\frac{b_0z^m+b_1z^{m-1}+\cdots+b_m}{a_0z^n+a_1z^{n-1}+\cdots+a_n}$$

在上式中，若 $m>n$，则会出现有 z 的正幂项，这在物理的可实现性上是不允许的；若 $m\leqslant n$，则不会出现有 z 的正幂项，$D(z)$ 物理上是可实现的，也即当 $m\leqslant n$ 时，产生 k 时刻的控制量 $u(k)$，最多只能利用直到 k 时刻的误差 $e(k)$、$e(k-1)$、…以及过去已知的控制量 $u(k-1)$、$u(k-2)$、…。

如果被控对象 $G(z)$ 含有 d 个采样周期的纯滞后 z^{-d}，根据式(5.72) $D(z)=\frac{1}{G(z)}\frac{\Phi(z)}{1-\Phi(z)}=\frac{\Phi(z)}{G(z)\Phi_e(z)}$，求取 $D(z)$ 可知，$D(z)$ 中将含有 z^d 的因子，则表示数字控制器具有超前特性，即在环节施加输入信号之前的 d 个采样周期系统就有输出，这样的超前环节即系统的响应超前于被控对象的输入在实际的物理上是不可实现的。所以为了物理的可实现性控制，所设计的闭环脉冲传递函数 $\Phi(z)$ 中必须包含 $G(z)$ 中的全部延迟因子 z^{-d}，即要把 $G(z)$ 中的纯滞后环节保留下来，以抵消 $G(z)$ 中的 z^{-d} 因子，避免 $D(z)$ 中存在超前环节 z^d。

3）最少拍控制器的稳定性要求

在前面最少拍系统控制器 $D(z)$ 的设计过程中，对由零阶保持器和被控对象组成的广义对象的脉冲传递函数 $G(z)$ 并未提出具体限制条件。实际上，只有广义对象的脉冲传递函数 $G(z)$ 是稳定的，即在单位圆外和圆上没有零极点，而且不含有纯滞后环节 z^{-d} 时，按前面的设计过程设计的最少拍系统才是稳定的、理想的。如果上述条件不能满足，就需要对设计原则做相应的调整和限制。

由式(5.70)可知

$$\Phi(z)=\frac{Y(z)}{R(z)}=\frac{D(z)G(z)}{1+D(z)G(z)}$$

可以看出，在系统的闭环脉冲传递函数中，$D(z)$ 和 $G(z)$ 总是成对出现的，但是不能简单地用 $D(z)$ 的极点和零点去抵消 $G(z)$ 中的单位圆上和圆外的零点和极点。这是因为简单地利用它们的零极点相互抵消，虽然理论上来说是可以得到一个稳定的闭环控制系统，但是这种稳定是建立在它们的零极点准确抵消的基础上的。在实际的控制中，由于存在对系统参数辨识的误差以及参数受外界环境影响产生漂移，并且由于计算机字长有限，$D(z)$ 难以精确表达，所以这种零极点的绝对抵消是不可能准确实现的，从而闭环系统并不真正稳定。

上述分析说明在单位圆外或圆上 $D(z)$ 和 $G(z)$ 不能对消零极点，但是这并不意味着含有这种现象的系统不能补偿成稳定的系统，只是在选择系统的闭环脉冲传递函数 $\Phi(z)$ 时必须增加约束条件，这种约束条件称为稳定性要求或稳定性约束条件，在设计系统时必须考虑。

由式(5.72) $D(z)=\frac{1}{G(z)}\frac{\Phi(z)}{1-\Phi(z)}=\frac{\Phi(z)}{G(z)\Phi_e(z)}$ 可知 $\Phi(z)=D(z)G(z)\Phi_e(z)$，根据

$\Phi(z)$的稳定性条件，$G(z)$的不稳定极点因为不能用$D(z)$去抵消，所以只能用$\Phi_e(z)$的不稳定零点来抵消。由表达式$\Phi(z)=1-\Phi_e(z)$，可知$\Phi(z)$和$\Phi_e(z)$有着相同的极点，稳定性要求$\Phi(z)$不能含有不稳定的极点，所以$\Phi_e(z)$也不能含有不稳定的极点。同时稳定性还要求$D(z)$不能含有不稳定的极点，所以$G(z)$的不稳定零点不能用$\Phi_e(z)$的不稳定的极点抵消，只能由$\Phi(z)$的不稳定零点来抵消。

根据以上推导可知要避免$G(z)$中单位圆上或圆外的零极点与$D(z)$的零极点相互抵消，则必须有以下两点。

(1) 当$G(z)$有单位圆上或圆外的零点时，在$\Phi(z)$的表达式中应把这些零点作为其零点而保留。

(2) 当$G(z)$有单位圆上或圆外的极点时，在$\Phi_e(z)$的表达式中应把这些极点作为其零点而保留。

这也是最少拍系统设计的稳定性要求或叫稳定性约束条件。

4. 任意广义对象的最少拍有纹波控制器的设计

最少拍有纹波控制器的设计是在最少拍控制器设计(只考虑快速性和准确性)的基础上考虑了系统的可实现性和稳定性的要求而得到的，也即必须考虑以下几个条件。

(1) 为了实现无静差调节，即准确性要求，在选择$\Phi_e(z)$时，必须针对不同的输入选择不同的形式，通用式为$\Phi_e(z)=(1-z^{-1})^q F(z)$。

(2) 为保证系统的稳定性，$\Phi_e(z)$的零点应包含抵消$G(z)$在单位圆上或圆外的不稳定极点，$\Phi(z)$的零点应包含抵消$G(z)$在单位圆上或圆外的所有零点。

(3) 为保证控制器$D(z)$的物理上可实现性，$\Phi(z)$应包含被控对象的纯滞后环节。

(4) 为了实现最少拍可知，即快速性要求，$F(z)$的形式应尽可能简单。即$F(z)$的选择要满足恒等式$\Phi(z)+\Phi_e(z)=1$。

假设被控对象的传递函数为$G_c(s)=G_c'(s)\mathrm{e}^{-\tau s}$，其中$G_c'(s)$是不含纯滞后部分的传递函数，$\tau$为纯滞后时间。则广义对象的脉冲传递函数$G(z)$为

$$\begin{aligned}G(z)&=z\left[\frac{1-\mathrm{e}^{-Ts}}{s}G_c(s)\right]\\&=Z\left[\frac{1-\mathrm{e}^{-Ts}}{s}G_c'(s)\mathrm{e}^{-\tau s}\right]\\&=z^{-\frac{\tau}{T}}\cdot Z\left[\frac{1-\mathrm{e}^{-Ts}}{s}G_c'(s)\right]\\&=z^{-d}\frac{B(z)}{A(z)}\end{aligned}\tag{5.88}$$

式中，设$d=\dfrac{\tau}{T}$，当$d=0$时被控对象不含纯滞后环节，当$d>0$时，被控对象含有d个采样周期的纯滞后。

假设$G(z)$含有u个在z平面的单位圆外或单位圆上的零点b_1、b_2、…、b_u；v个在z平面的单位圆上或圆外的极点a_1、a_2、…、a_v，其中j个极点在单位圆上；$G'(z)$是$G(z)$中不含单位圆上或圆外的零极点部分，那么广义对象的传递函数可表示为

$$G(z)=z^{-d}\frac{\prod\limits_{i=1}^{u}(1-b_iz^{-1})}{\prod\limits_{i=1}^{v-j}(1-a_iz^{-1})(1-z^{-1})^j}G'(z)\tag{5.89}$$

其中$|b_i|\geqslant 1$，$|a_i|>1$，表明是在单位圆外的零极点。

由分析可知，选择系统的闭环脉冲传递函数必须满足以下约束条件。

（1）在$\Phi_e(z)$的零点中，必须包含$G(z)$在z平面单位圆外或单位圆上的所有极点，即：

$$\Phi_e(z)=1-\Phi(z)=\left[\prod_{i=1}^{v-j}(1-a_i z^{-1})\right](1-z^{-1})^j F_1(z) \tag{5.90}$$

其中，$F_1(z)$是关于z^{-1}的多项式，且不包含$G(z)$中的不稳定极点a_i。为了使$\Phi_e(z)$能够实现，$F_1(z)$的形式应为

$$F_1(z)=1+f_{11}z^{-1}+f_{12}z^{-2}+\cdots+f_{1m}z^{-m} \tag{5.91}$$

而根据前面最少拍控制的含义确定可知$\Phi_e(z)=(1-z^{-1})^q$，因此综合最少拍控制的含义和系统的稳定性要求可得

$$\Phi_e(z)=1-\Phi(z)=\left[\prod_{i=1}^{v-j}(1-a_i z^{-1})\right](1-z^{-1})^k F_1(z) \tag{5.92}$$

式中，$k=\max(j,q)$。由式(5.92)可知：

① 若$j\leqslant q$，则$\Phi_e(z)=1-\Phi(z)=\left[\prod_{i=1}^{v-j}(1-a_i z^{-1})\right](1-z^{-1})^q F_1(z)$，其中$z^{-1}$的阶数为$v-j+q+m$。

② 若$j>q$，则$\Phi_e(z)=1-\Phi(z)=\left[\prod_{i=1}^{v-j}(1-a_i z^{-1})\right](1-z^{-1})^j F_1(z)$，其中$z^{-1}$的阶数为$v+m$。

（2）在$\Phi(z)$的零点中，必须包含$G(z)$在z平面单位圆外或单位圆上的所有零点，并包含纯滞后环节z^{-d}，即：

$$\Phi(z)=z^{-d}\left[\prod_{i=1}^{u}(1-b_i z^{-1})\right]F_2(z) \tag{5.93}$$

其中，$F_2(z)$是关于z^{-1}的多项式，且不包含$G(z)$中的不稳定零点b_i。为了使$\Phi(z)$能够实现，$F_2(z)$的形式应为

$$F_2(z)=f_{21}z^{-1}+f_{22}z^{-2}+\cdots+f_{2n}z^{-n} \tag{5.94}$$

其中z^{-1}的阶数为$d+u+n$。

（3）$F_1(z)$和$F_2(z)$阶数的确定满足以下两条原则。

① 因为$\Phi_e(z)=1-\Phi(z)$，所以选择的阶数应使等式两边的最高幂次相同，而且同幂次项的系数也要相等，所以通过待定系数法确定$\Phi(z)$和$\Phi_e(z)$的系数，从而最终确定$\Phi(z)$和$\Phi_e(z)$的数学表达式。

② 选择的$F_1(z)$和$F_2(z)$应使$\Phi(z)$的阶次为最低，也即有：

若$G(z)$有j个极点在单位圆上，当$j\leqslant q$时，有$\begin{cases}m=u+d\\ n=v-j+q\end{cases}$，

若$G(z)$有j个极点在单位圆上，当$j>q$时，有$\begin{cases}m=u+d\\ n=v\end{cases}$。

根据以上确定$\Phi(z)$和$\Phi_e(z)$时必须满足的约束条件，从而可得到最少拍控制器为

$$D(z)=\frac{1}{G(z)}\frac{\Phi(z)}{1-\Phi(z)}=\begin{cases}\dfrac{F_2(z)}{G'(z)(1-z^{-1})^{q-j}F_1(z)}, & j\leqslant q\\[2ex] \dfrac{F_2(z)}{G'(z)F_1(z)}, & j>q\end{cases} \tag{5.95}$$

根据以上约束条件设计出来的最少拍控制系统，只保证了在最少的几个采样周期后，系统的输出响应在采样点的误差为零，而并不能保证系统的输出在任意两个采样点之间的误差为零，采样点之间的偏差通常称之为纹波，也就是说非采样点有纹波存在。这样设计出来的控制系统的输出信号 $y(t)$ 有纹波存在，故称其为最少拍有纹波控制系统，此种控制器称为最少拍有纹波控制器。

这种控制系统的输出信号 $y(t)$ 的纹波在采样点上观测不到，要用修正 Z 变换方能计算得出两个采样点之间的输出值，这种纹波称为隐蔽振荡(Hidden Oscillations)。

【例 5.14】 如图 5.16 所示的计算机控制系统中，假设被控对象的传递函数为 $G_c(s)=\dfrac{10}{s(T_m s+1)}$。已知 $T=T_m=0.025\text{s}$，试针对单位速度输入函数，设计最少拍有纹波系统，并画出数字控制器和系统的输出波形。

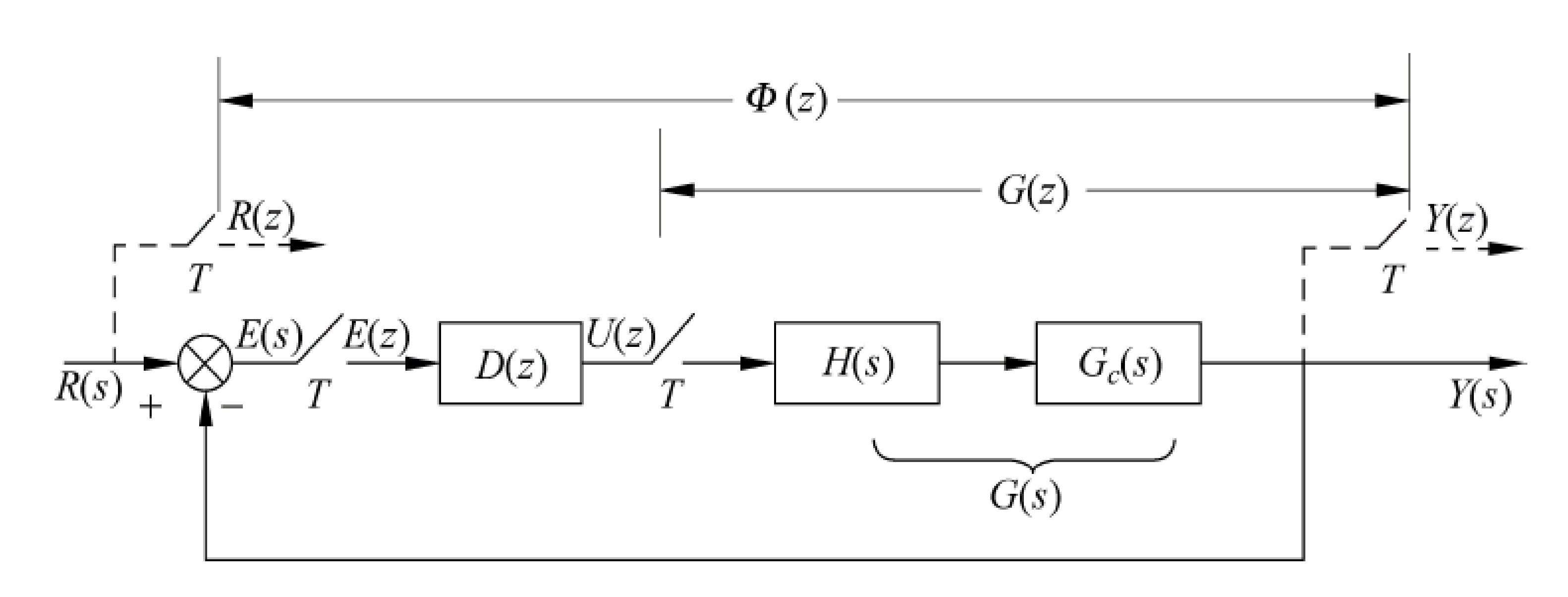


图 5.16 计算机控制系统方框图

解：由题意可求得广义对象的传递函数 $G(s)$ 为

$$G(s)=H(s)G_c(s)=\frac{1-e^{-Ts}}{s}\,\frac{10}{s(T_m s+1)}$$

将其展开可求得其脉冲传递函数 $G(z)$ 为

$$\begin{aligned}G(z)&=Z[G(s)]=10(1-e^{-Ts})\left[\frac{1}{s^2}-\frac{T_m}{s}+\frac{T_m^2}{T_m s+1}\right]\\&=10(1-z^{-1})\left[\frac{Tz^{-1}}{(1-z^{-1})^2}-\frac{T_m}{1-z^{-1}}+\frac{T_m}{1-e^{-T/T_m}z^{-1}}\right]\end{aligned}$$

代入 $T=T_m=0.025\text{s}$，化简可得到

$$G(z)=\frac{0.092z^{-1}(1+0.718z^{-1})}{(1-z^{-1})(1-0.368z^{-1})}$$

由 $G(z)$ 的表达式可以看出其零点为 -0.718(单位圆内)，极点为 1(单位圆上)和 0.368(单位圆内)，因此可知 $d=0$(注意分子中的 z^{-1} 不能当作滞后因子)，$u=0$，$v=1$，$j=1$，$q=2$，且 $j<q$，则有 $m=u+d=0$；$n=v-j+q=2$。

对于单位速度输入进行设计，选择

$$\Phi_e(z)=1-\Phi(z)=\left[\prod_{i=1}^{v-j}(1-a_i z^{-1})\right](1-z^{-1})^q F_1(z)=(1-z^{-1})^2$$

$$\Phi(z)=z^{-d}\left[\prod_{i=1}^{u}(1-b_i z^{-1})\right]F_2(z)=f_{21}z^{-1}+f_{22}z^{-2}$$

由 $\Phi_e(z)=1-\Phi(z)$，可得 $1-f_{21}z^{-1}-f_{22}z^{-2}=(1-z^{-1})^2$，根据多项式相等，其对应系

数相等的性质，有 $f_{21}=2, f_{22}=-1$，所以其系统闭环脉冲传递函数为 $\Phi(z)=2z^{-1}-z^{-2}$。

所以其数字控制器的脉冲传递函数为

$$D(z)=\frac{1}{G(z)}\frac{\Phi(z)}{1-\Phi(z)}=\frac{21.8(1-0.5z^{-1})(1-0.368z^{-1})}{(1+0.718z^{-1})(1-z^{-1})}$$

进一步可求得

$$E(z)=R(z)\Phi_e(z)=\frac{Tz^{-1}}{(1-z^{-1})^2}(1-z^{-1})^2=Tz^{-1}=0.025z^{-1}$$

$$Y(z)=R(z)\Phi(z)=\frac{Tz^{-1}}{(1-z^{-1})^2}(2z^{-1}-z^{-2})=T(2z^{-2}+3z^{-3}+4z^{-4}+\cdots)$$

$$U(z)=E(z)D(z)=0.025z^{-1}\frac{21.8(1-0.5z^{-1})(1-0.368z^{-1})}{(1+0.718z^{-1})(1-z^{-1})}$$

$$=0.54z^{-1}-0.32z^{-2}+0.40z^{-3}-0.12z^{-4}+0.25z^{-5}+\cdots$$

由以上可画出有纹波系统数字控制器和系统的输出波形如图 5.17(a)和图 5.17(b)所示。

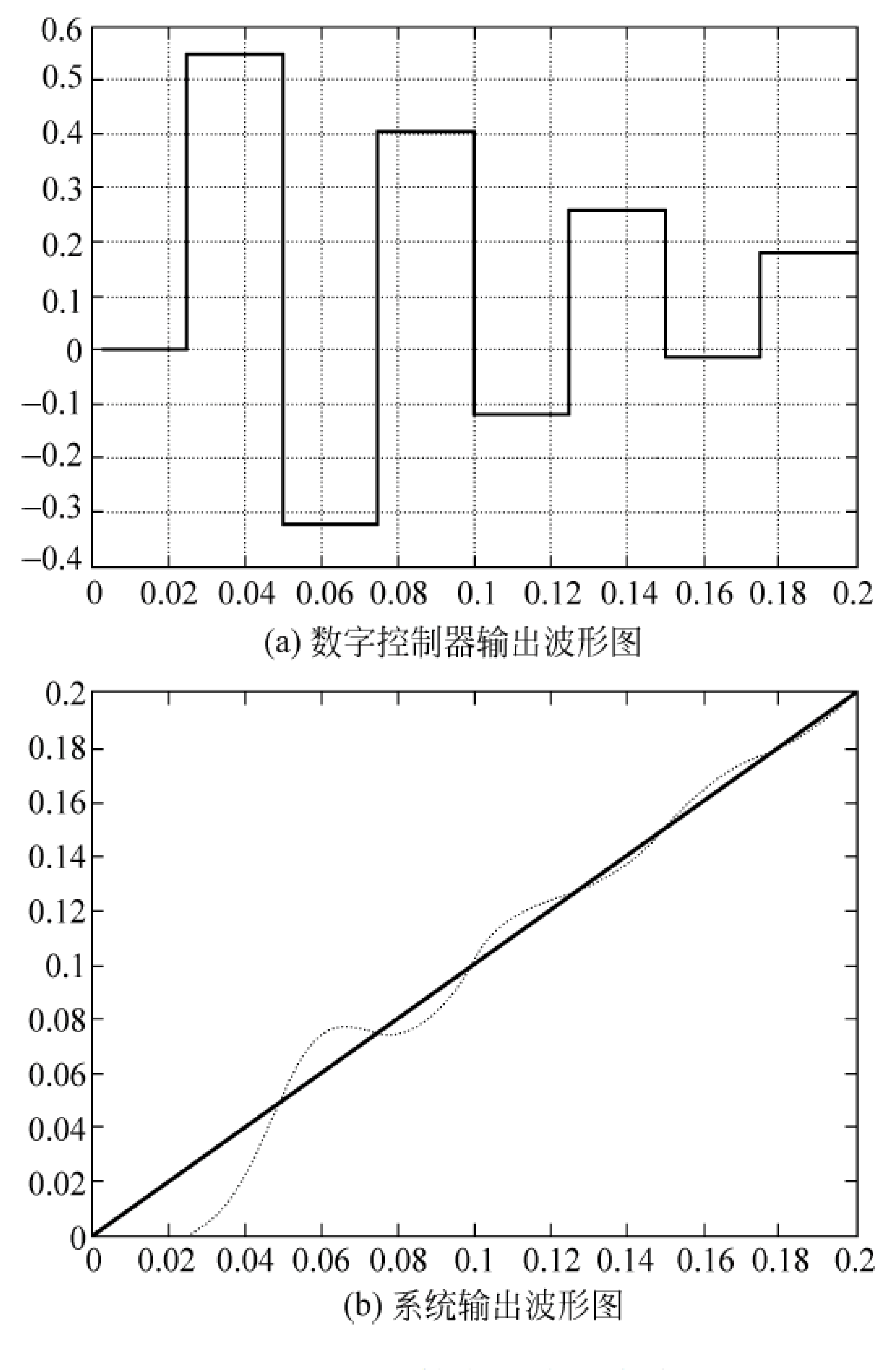


图 5.17　最少拍有纹波系统波形

由图 5.17 可以看出，系统对于单位速度输入，经过两拍后系统的输出在采样点上能跟踪输入的值，但在采样点之间，系统的输出与输入给定值不一致，有纹波存在。

5. 最少拍无纹波控制器的设计

按最少拍有纹波系统设计的控制器，其系统的输出响应只保证了在采样点上的误差为零，而不能保证任意两个采样点之间的误差也为零。其原因在于数字控制器的输出序列 $u(k)$ 在经过若干拍后，不为常数或零，而是振荡收敛的。这种非采样时刻纹波的存在不仅造成非采样时刻的误差，而且也浪费执行机构的功率，消耗能量，同时增加机械磨损，因此在设计时应加以考虑消除这种纹波。

最少拍无纹波数字控制器的设计思想是：系统在典型的输入信号作用下，经过尽可能少的采样周期后，输出达到稳态，在采样点上及采样点之间均能保证误差为零。这是以牺牲快速性为代价的，它的调节时间一般要长于有纹波系统。下面介绍最少拍无纹波控制器的设计方法。

1）最少拍无纹波控制器实现的必要条件

若要在稳态过程中获得无纹波的输出，即输出信号在采样点之间无纹波，则必须使输出信号在两采样点之间能跟随输入信号，即要满足以下条件。

（1）对阶跃输入，当 $t \geqslant NT$ 时，有 $y(t)=$ 常数。

（2）对速度输入，当 $t \geqslant NT$ 时，有 $y'(t)=$ 常数。

（3）对加速度输入，当 $t \geqslant NT$ 时，有 $y''(t)=$ 常数。

这样要求被控对象 $G_c(s)$ 必须有能力给出与系统输入 $r(t)$ 相同的、平滑的输出 $y(t)$。例如针对单位速度输入函数设计最少拍无纹波系统，被控对象 $G_c(s)$ 的输入 $u(t)$ 是由零阶保持器重构的，而最少拍无纹波系统 $u(t)$ 的值在两采样点之间是不变的，即保持为常数输入到被控对象上。因此为了使被控对象产生和系统输入同样的速度函数，则被控对象 $G_c(s)$ 中必须至少含有一个积分环节，这样使得在常值的控制信号作用下，其稳态输出也是所要求的单位速度输出。同样，针对加速度输入函数设计的无纹波控制器，则被控对象 $G_c(s)$ 中必须至少含有两个积分环节。

被控对象必须具有足够的积分环节是实现无纹波控制的必要条件。这样以保证 $u(t)$ 为常数时，$G_c(s)$ 的稳态输出能完全跟踪输入，且无纹波。

2）最少拍无纹波中确定闭环脉冲传递函数 $\Phi(z)$ 的约束条件

在前面的分析中可知如果系统进入稳态后，输入到被控对象 $G_c(s)$ 的控制信号 $u(t)$ 还有波动，则稳态过程中系统输出就会有纹波。因此要使系统在稳态过程中无纹波，就要求稳态时的控制信号 $u(t)$ 为常数，或者为零。数字控制器的输出信号 $u(t)$ 的 Z 变换幂级数展开式为

$$U(z) = \sum_{k=0}^{\infty} u(k) z^{-k} = u(0) + u(1) z^{-1} + \cdots + u(l) z^{-l} + u(l+1) z^{-(l+1)} + \cdots \quad (5.96)$$

如果系统经过 l 个采样周期到达稳态，无纹波就要求 $u(l)=u(l+1)=u(l+2)=\cdots=$ 常数或者零。

设广义对象的脉冲传递函数 $G(z)$ 为

$$G(z) = z^{-d} \frac{B(z)}{A(z)} \quad (5.97)$$

则由 $Y(z)=G(z)U(z)$ 和 $Y(z)=\Phi(z)R(z)$ 可知

$$U(z)=\frac{Y(z)}{G(z)}=\frac{\Phi(z)}{G(z)}R(z)=\frac{\Phi(z)}{z^{-d}B(z)}A(z)R(z) \tag{5.98}$$

也即 $\Phi_u(z)=\frac{U(z)}{R(z)}=\frac{\Phi(z)}{z^{-d}B(z)}A(z)$。

要使控制信号 $u(t)$ 在稳态过程中为零或者常数，那么它的 Z 变换 $U(z)$ 对输入 $R(z)$ 的脉冲传递函数之比 $\Phi_u(z)$ 只能是关于 z^{-1} 的有限项多项式。因此由式(5.98)可知闭环脉冲传递函数 $\Phi(z)$ 必须包含 $G(z)$ 的分子多项式 $B(z)$ 以及滞后环节，也即所有的零点和滞后环节，这样才能保证分式能整除。因此原来最少拍有纹波设计时确定 $\Phi(z)$ 的式(5.96)应修改为

$$\Phi(z)=z^{-d}B(z)F_2(z)=z^{-d}\left[\prod_{i=1}^{w}(1-b_iz^{-1})\right]F_2(z) \tag{5.99}$$

式中，w 为 $G(z)$ 所有零点(包括单位圆内、单位圆上以及单位圆外的零点)数，b_1、b_2、…、b_w 为其所有零点，$F_2(z)=f_{21}z^{-1}+f_{22}z^{-2}+\cdots+f_{2n}z^{-n}$。

由以上推导可以看出有纹波和无纹波设计的唯一区别在于：有纹波设计时，$\Phi(z)$ 包含 $G(z)$ 单位圆上和单位圆外的零点；无波纹设计时，$\Phi(z)$ 包含 $G(z)$ 单位圆内、单位圆上和单位圆外的所有零点。这样最少拍无纹波系统阶数要比最少拍有纹波系统的阶数要高，所以无纹波系统的调整时间要增加若干拍，增加的拍数等于 $G(z)$ 在单位圆内的零点数。

3) 最少拍无纹波中确定闭环脉冲传递函数 $\Phi(z)$ 的方法

确定 $\Phi(z)$ 时必须满足下列条件。

(1) 被控对象中含有足够的积分环节，以满足无纹波系统设计的必要条件。

(2) 按式(5.99) $\Phi(z)=z^{-d}\left[\prod_{i=1}^{w}(1-b_iz^{-1})\right]F_2(z)$ 来选择 $\Phi(z)$。

(3) 按式 $\Phi_e(z)=1-\Phi(z)=\left[\prod_{i=1}^{v-j}(1-a_iz^{-1})\right](1-z^{-1})^qF_1(z),q\geqslant j$ 和 $\Phi_e(z)=1-\Phi(z)=\left[\prod_{i=1}^{v-j}(1-a_iz^{-1})\right](1-z^{-1})^jF_1(z),q<j$ 选择 $\Phi_e(z)$。

(4) $F_1(z)=1+f_{11}z^{-1}+f_{12}z^{-2}+\cdots+f_{1m}z^{-m}$ 和 $F_2(z)=f_{21}z^{-1}+f_{22}z^{-2}+\cdots+f_{2n}z^{-n}$ 阶数的确定可按以下方法来选取，即

若 $G(z)$ 有 j 个极点在单位圆上，当 $j\leqslant q$ 时，有 $\begin{cases}m=w+d\\n=v-j+q\end{cases}$，

若 $G(z)$ 有 j 个极点在单位圆上，当 $j>q$ 时，有 $\begin{cases}m=w+d\\n=v\end{cases}$。

【例 5.15】 在图 5.16 所示的计算机控制系统中，假设被控对象的传递函数为 $G_c(s)=\frac{10}{s(T_ms+1)}$。已知 $T=T_m=0.025\text{s}$，试针对单位速度输入函数，设计最少拍无纹波系统，并绘出数字控制器和系统的输出波形图。

解：由 $G_c(s)$ 的表达式可知其中有一个积分环节，说明它有能力平滑地产生等速输出响应，满足无纹波系统设计的必要条件。

由例 5.14 可知零阶保持器和被控对象组成的广义对象的脉冲传递传递函数 $G(z)$ 为

$$G(z)=\frac{0.092z^{-1}(1+0.718z^{-1})}{(1-z^{-1})(1-0.368z^{-1})}$$

由 $G(z)$ 的表达式可以看出其零点为 -0.718（单位圆内），极点为 1（单位圆上）和 0.368（单位圆内），因此可知 $d=0$（注意分子中的 z^{-1} 不能当作滞后因子），包含一个零点 $w=1$，包含一个单位圆上的极点 $v=1$，且一个极点在单位圆上 $j=1$，输入为单位速度函数 $q=2$，且 $j<q$，则有 $m=w+d=1$；$n=v-j+q=2$。所以 $F_1(z)=1+f_{11}z^{-1}$；$F_2(z)=f_{21}z^{-1}+f_{22}z^{-2}$。

对于单位速度输入进行设计，选择

$$\Phi_e(z)=1-\Phi(z)=\left[\prod_{i=1}^{v-j}(1-a_iz^{-1})\right](1-z^{-1})^qF_1(z)=(1-z^{-1})^2(1+f_{11}z^{-1})$$

$$\Phi(z)=z^{-d}\left[\prod_{i=1}^{w}(1-b_iz^{-1})\right]F_2(z)=(1+0.718z^{-1})(f_{21}z^{-1}+f_{22}z^{-2})$$

由 $\Phi_e(z)=1-\Phi(z)$，可得 $1-(1+0.718z^{-1})(f_{21}z^{-1}+f_{22}z^{-2})=(1-z^{-1})^2(1+f_{11}z^{-1})$，展开有 $1+(f_{11}-2)z^{-1}+(1-2f_{11})z^{-2}+f_{11}z^{-3}=1-f_{21}z^{-1}-(f_{22}+0.718f_{21})z^{-2}-0.718f_{22}z^{-3}$，根据多项式相等，其对应系数相等的性质，得：$f_{11}-2=-f_{21}$，$1-2f_{11}=-(f_{22}+0.718f_{21})$，$f_{11}=-0.718f_{22}$。解得：$f_{11}=0.592$，$f_{21}=1.408$，$f_{22}=-0.825$。故可得其系统的闭环脉冲传递函数为 $\Phi(z)=(1+0.718z^{-1})(1.408z^{-1}-0.825z^{-2})$，$\Phi_e(z)=(1-z^{-1})^2(1+0.592z^{-1})$。

所以其数字控制器的脉冲传递函数为

$$D(z)=\frac{1}{G(z)}\frac{\Phi(z)}{1-\Phi(z)}=\frac{15.29(1-0.587z^{-1})(1-0.368z^{-1})}{(1+0.592z^{-1})(1-z^{-1})}$$

进一步可求得闭环系统的输出序列为

$$\begin{aligned}Y(z)&=R(z)\Phi(z)=\frac{Tz^{-1}}{(1-z^{-1})^2}(1+0.718z^{-1})(1.408z^{-1}-0.825z^{-2})\\&=1.41Tz^{-2}+3Tz^{-3}+4Tz^{-4}+5Tz^{-5}+\cdots\end{aligned}$$

$$E(z)=R(z)\Phi_e(z)=\frac{Tz^{-1}}{(1-z^{-1})^2}(1-z^{-1})^2(1+0.592z^{-1})=T(z^{-1}+0.592z^{-2})$$

数字控制器的输出序列为

$$\begin{aligned}U(z)&=\frac{Y(z)}{G(z)}=\frac{R(z)\Phi(z)}{G(z)}\\&=\frac{Tz^{-1}}{(1-z^{-1})^2}(1+0.718z^{-1})(1.408z^{-1}-0.825z^{-2})\frac{(1-z^{-1})(1-0.368z^{-1})}{0.092z^{-1}(1+0.718z^{-1})}\\&=0.38z^{-1}+0.02z^{-2}+0.09z^{-3}+0.09z^{-4}+\cdots\end{aligned}$$

由此可得无纹波系统数字控制器和系统的输出波形如图 5.18(a)和图 5.18(b)所示。

由例 5.14 和例 5.15 的输出波形图可以看出，对于单位速度输入，有纹波系统的调整时间为两个采样周期，系统的输出跟随输入函数后，由于数字控制器的输出仍在波动，所以系统的输出在非采样时刻有纹波。无纹波系统的调整时间为三个采样周期，系统输出跟随输入函数所需的时间比有纹波系统增加了一拍，因为 $G(z)$ 在单位圆内有一个零点。同时由于系统中数字控制器的输出经过三个采样周期后为常值，所以无纹波系统在采样点之间不存在纹波。

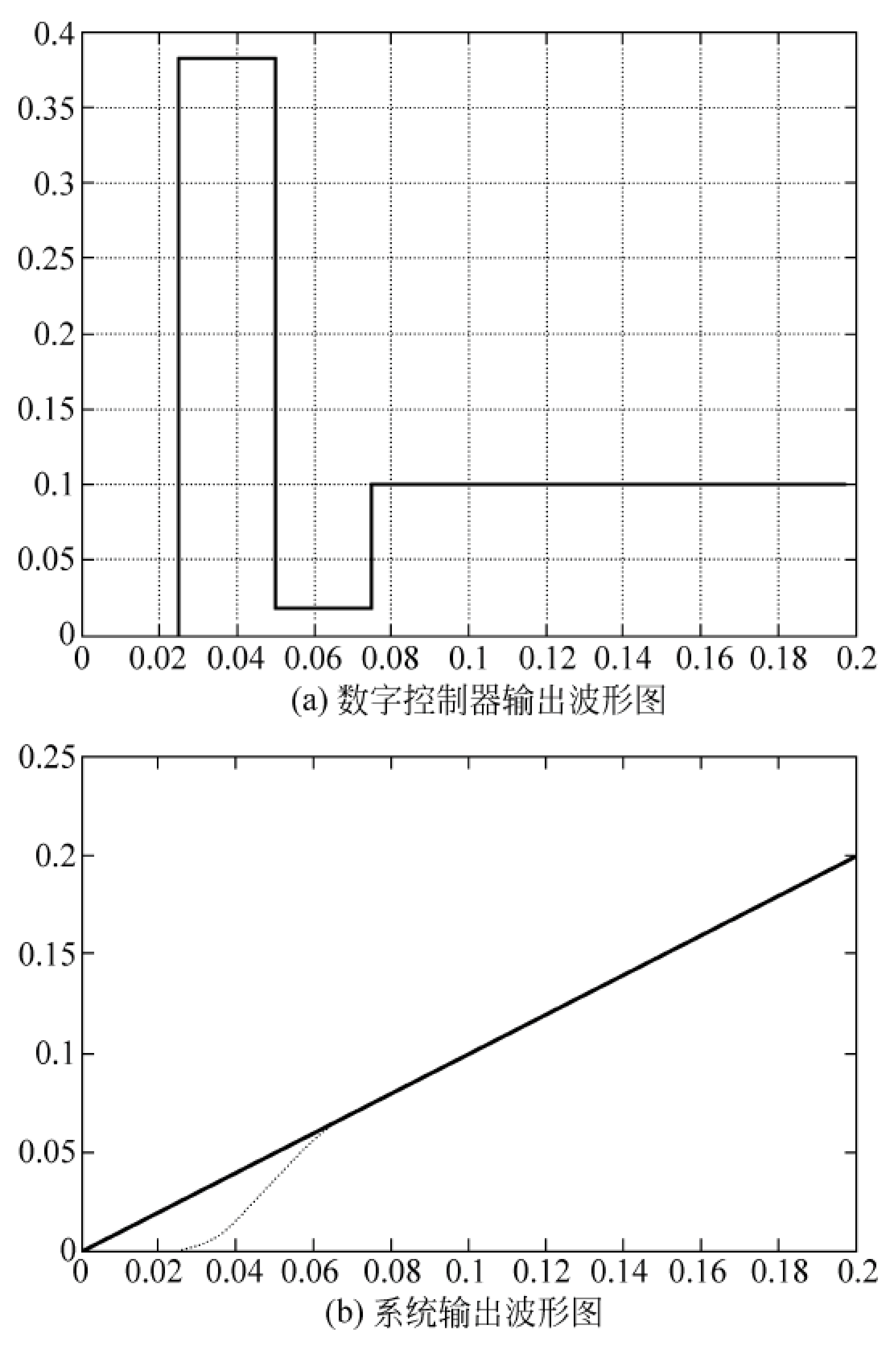


图 5.18 最少拍无纹波系统波形

5.5 史密斯预估控制

前面讨论的学习方法是针对单回路、单变量和控制规律比较简单的控制系统的设计，除此之外计算机控制系统中还有比较复杂的控制规律，比如纯滞后补偿控制、串级控制、前馈控制以及多变量控制等。

在一些工业生产过程(如热工、化工)中，由于能量或物料的传输延迟，许多被控对象含有较大的纯滞后特性。对象的纯滞后特性会使系统的稳定性降低，动态特性变坏，容易引起系统超调和持续振荡。具有纯滞后特性的对象属于比较难于控制的一类对象，采用常规的PID控制器很难获得满意的控制效果。早在20世纪50年代，国外就对工业生产过程中的纯滞后对象进行了深入的研究。对这一类对象，一般来说快速性要求是次要的，而对稳定性、不产生超调的要求是主要的。基于此，人们提出了多种设计方法，比较有代表性的方法有纯滞后补偿控制，主要是美国IBM公司的Dahlin在1968年提出的大林算法和史密斯(Smith)在1957年提出的Smith补偿法。Smith补偿法由于模拟仪表不能实现这种补偿，致使这种方法在工程中无法实现，现在利用微型计算机在计算机控制系统中能够方便地实现，这里主要介绍Smith补偿法。

5.5.1 史密斯预估控制原理

在图 5.19 所示的单回路控制系统中，控制器的传递函数为 $D(s)$，被控对象传递函数为 $G_P(s)\mathrm{e}^{-\tau s}$，被控对象中不包含纯滞后部分的传递函数为 $G_P(s)$，τ 为纯滞后的时间，被控对象纯滞后部分的传递函数为 $\mathrm{e}^{-\tau s}$。

图 5.19 带纯滞后环节的控制系统

对于图 5.19 所示的系统的闭环传递函数为

$$\Phi(s)=\frac{D(s)G_P(s)\mathrm{e}^{-\tau s}}{1+D(s)G_P(s)\mathrm{e}^{-\tau s}} \tag{5.100}$$

由式(5.100)可以看出，闭环传递函数分母也即系统特征方程 $1+D(s)G_P(s)\mathrm{e}^{-\tau s}=0$ 中包含纯滞后环节 $\mathrm{e}^{-\tau s}$，它会降低系统的稳定性，如果 τ 较大时，系统甚至可能变得不稳定。因此常规的调节规律 $D(s)$ 很难使具有纯滞后的系统获得满意的控制性能。

为了改善控制系统的性能，引入史密斯预估器将对象进行改造。史密斯预估器的原理是：与控制器 $D(s)$ 并接一个补偿环节，用来补偿被控对象中的纯滞后部分，这个补偿环节称为预估器，其传递函数为 $G_P(s)(1-\mathrm{e}^{-\tau s})$，$\tau$ 为纯滞后时间，补偿后的系统框图如图 5.20 所示。

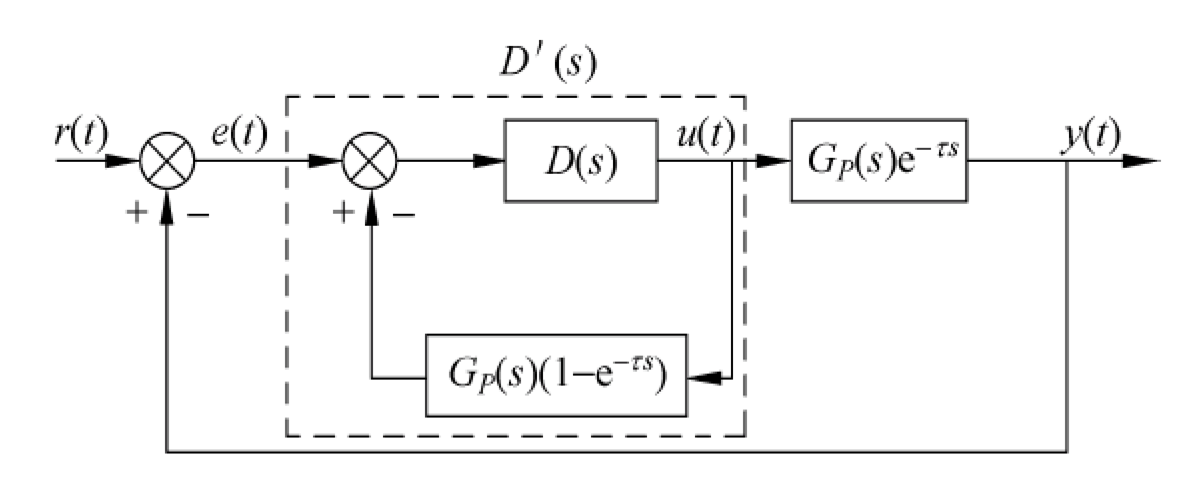


图 5.20 史密斯补偿后的控制系统

图 5.20 中的虚框即由控制器 $D(s)$ 和史密斯预估器组成的补偿回路，称为纯滞后补偿器，其传递函数 $D'(s)$ 为

$$D'(s)=\frac{D(s)}{1+D(s)G_P(s)(1-\mathrm{e}^{-\tau s})} \tag{5.101}$$

所以其经过补偿后系统的闭环传递函数为

$$\Phi(s)=\frac{D'(s)G_P(s)\mathrm{e}^{-\tau s}}{1+D'(s)G_P(s)\mathrm{e}^{-\tau s}}=\frac{D(s)G_P(s)}{1+D(s)G_P(s)}\cdot \mathrm{e}^{-\tau s} \tag{5.102}$$

由式(5.102)可以看出，经过史密斯补偿后，闭环系统的特征方程为 $1+D(s)G_P(s)=0$，其中已不包含纯滞后环节，消除了纯滞后的影响。因为纯滞后环节 $\mathrm{e}^{-\tau s}$ 已在闭环控制回路之外，它不会影响系统的稳定性。拉普拉斯变换的位移定理说明 $\mathrm{e}^{-\tau s}$ 仅仅将控制作用在时间坐标上推移了一个时间 τ，而控制系统的过渡过程及其他性能指标都与被控对象特性为 $G_P(s)$(即没有纯滞后)时完全相同。因此将史密斯预估器与控制器并联，理论上可以使控制对象的时间滞后得到完全补偿。

5.5.2 具有纯滞后补偿的数字控制器

由图 5.20 可以得到带有史密斯预估器的计算机控制系统结构框图，如图 5.21 所示，图

中，$H_0(s)$为零阶保持器，$D(z)$为数字控制器脉冲传递函数，可采用数字 PID 控制器，也可以采用其他控制算法来实现，$G_P(z)$为被控对象中不具有纯滞后部分的脉冲传递函数，z^{-N}为被控对象纯滞后部分的脉冲传递函数，$N=\tau/T$，τ 是被控对象的纯滞后时间，T 是系统的采样周期。纯滞后补偿的数字控制器如图 5.21 中的虚线框部分，它由两部分组成：一部分是常规的数字控制器的脉冲传递函数 $D(z)$，可由 $D(s)$ 离散化得到，另一部分是史密斯预估器，这两部分均用计算机来实现。

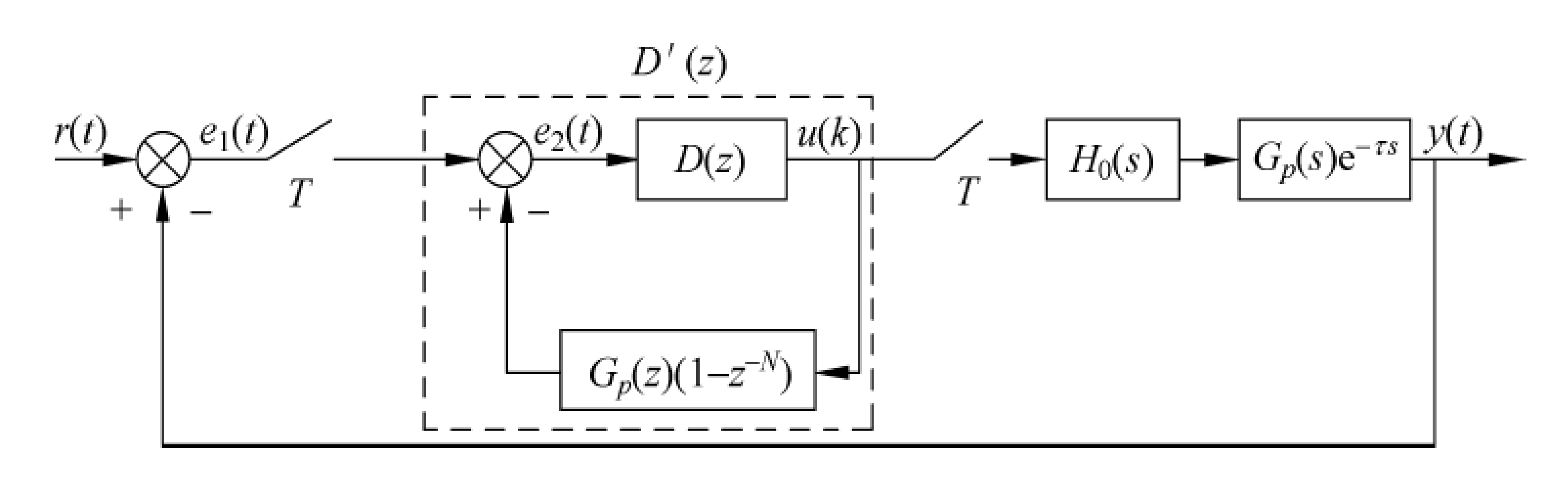


图 5.21 带史密斯预估器的计算机控制系统

采用模拟调节仪表来实现上述史密斯预估器比较困难，这是因为不同的对象模型 $G_P(s)$ 各式各样，纯滞后环节由模拟电路来模拟实现也很麻烦，但相反地由计算机来实现史密斯预估器却很容易。

下面以一阶惯性纯滞后对象为例说明史密斯纯滞后补偿器的计算机实现过程。许多工业对象可近似用一阶惯性环节和纯滞后环节的串联来表示，即

$$G_c(s)=G_P(s)\mathrm{e}^{-\tau s}=\frac{K_f}{T_f s+1}\mathrm{e}^{-\tau s}$$

式中，K_f为被控对象的放大倍数；T_f为被控对象的时间常数；τ 是被控对象的纯滞后时间。为了便于说明史密斯补偿器的计算机实现过程，将图 5.21 中的虚线框部分中的史密斯预估器可以变换为图 5.22 所示的形式。

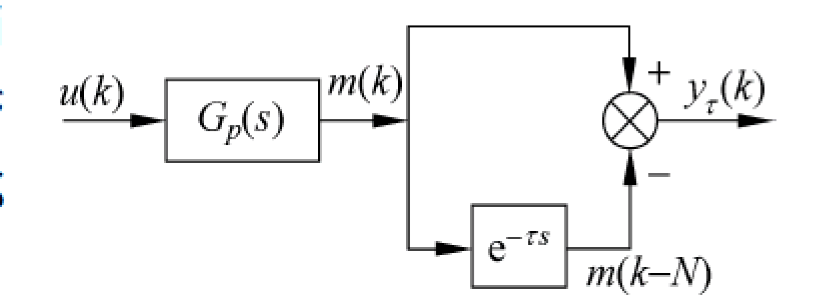


图 5.22 史密斯预估器方框图

令 $N=\tau/T$（取整数），则数字史密斯补偿器的输出为

$$y_\tau(k)=m(k)-m(k-N) \tag{5.103}$$

式中，$m(k)$为中间变量，其算式与对象模型有关，对象的模型不同，$m(k)$的算式也是不同的。这样就有$\dfrac{M(s)}{U(s)}=G_P(s)=\dfrac{K_f}{T_f s+1}$，其相应的微分方程为

$$T_f\frac{\mathrm{d}m(t)}{\mathrm{d}t}+m(t)=K_f u(t) \tag{5.104}$$

对上式进行离散化处理的

$$T_f\frac{m(k)-m(k-1)}{T}+m(k)=K_f u(k) \tag{5.105}$$

整理式(5.105)后可得

$$m(k)=am(k-1)+bu(k) \tag{5.106}$$

式中，$a=\dfrac{T_f}{T_f+T}$，$b=\dfrac{k_f T}{T_f+T}$。

由以上推导可以看出史密斯补偿器的差分方程中有 $m(k-N)$项，可以用存储单元法来

产生纯滞后信号。即为了形成纯滞后的 N 步信号，需要在计算机内存中开辟 $N+1$ 个单元用来存放 $m(k)$ 的历史数据。实施时，在每个采样周期，先将各存储单元的内容顺序移至下一个存储单元，即把新得到的 $m(k)$ 存入 0 号单元，同时把 0 号单元原来存放的数据移到 1 号单元，1 号单元原来存放的数据移到 2 号单元，依次类推，N 号单元里的内容即为 $m(k)$ 滞后 N 个采样周期后的信号 $m(k-N)$，其结构如图 5.23 所示。这种方法精度高，但占用一定的内存，且 N 越大，占用的内存量也越大。

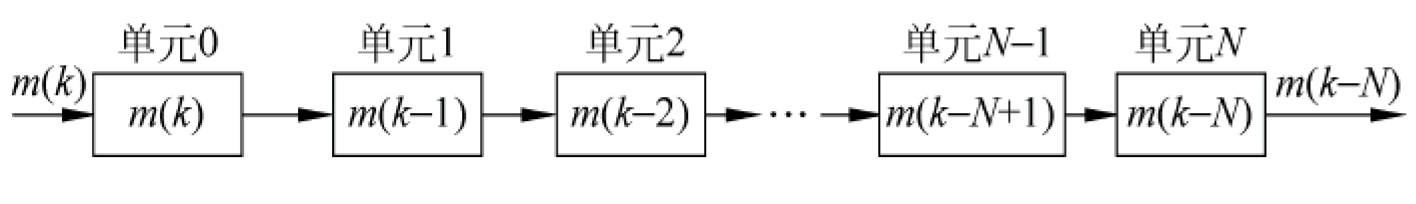


图 5.23 $m(k-N)$ 信号形成示意图

用上述方法产生纯滞后信号后，由式(5.103)即可求出 $y_\tau(k)$。所以史密斯纯滞后补偿控制算法的步骤为以下几项。

(1) 计算反馈回路的偏差 $e_1(k)=r(k)-y(k)$。

(2) 计算中间变量 $m(k)$，当对象为一阶惯性环节加纯滞后时，$m(k)$ 可按式(5.106)来计算。

(3) 求取 $m(k-N)$。

(4) 计算史密斯预估器的输出 $y_\tau(k)$，由式(5.103)即可求出 $y_\tau(k)=m(k)-m(k-N)$。

(5) 计算 $D(z)$ 的输入，也即偏差 $e_2(k)=e_1(k)-y_\tau(k)=r(k)-y(k)-y_\tau(k)$。

(6) 计算控制器的输出 $u(k)$。当控制器采用 PID 控制算法时，则有

$$u(k)=u(k-1)+\Delta u(k)=u(k-1)+K_P[e_2(k)-e_2(k-1)]$$
$$+K_I e_2(k)+K_D[e_2(k)-2e_2(k-1)+e_2(k-2)]$$

式中，K_P 为 PID 控制的比例系数，$K_I=K_P T/T_i$ 为积分系数，$K_D=K_P T_d/T$ 为微分系数。$u(k)$ 经过 D/A 输出后直接作用到执行机构上，就可以实现对被调量的纯滞后补偿控制。

一般认为，史密斯预估补偿方法是解决大滞后问题的有效方法，但实现史密斯预估控制的关键是对象要有精确的数学模型，在模型基本准确时能表现出良好的性能。因此对于一些复杂而难以用数学模型描述的系统，此方法则显得无能为力。

5.6 串级控制

串级控制是在单回路 PID 控制的基础上发展起来的一种控制技术，并且得到了广泛应用。一般情况下，单回路 PID 控制系统已经能满足工业过程对控制的要求，因此它也是一种应用最基本和最广泛的系统。但当系统中同时有几个干扰因素影响到同一被控量时，如果仍采用单回路控制系统来控制其中一个变量，则难以满足系统的控制性能要求。串级控制是在原来单回路控制的基础上，增加一个或几个控制内回路，用以控制可能引起被控量变化的其他因素，从而抑制被控对象的时滞特性，提高系统动态响应的快速性。

5.6.1 串级控制的结构和原理

一般串级控制系统结构框图如图 5.24 所示。从图中可以看出，通用的串级系统在结构

上形成了两个闭环。其中外面的闭环称为主回路或主环,在控制中用来完成“细调”的任务,用于最终保证被控量满足控制工艺要求;里面的闭环称为副回路或副环,主要用于克服被控对象所受到的主要干扰,在控制过程中起着“粗调”的作用。系统中有两个调节器,主调节器和副调节器,其中主调节器具有自己独立的给定值,其输出作为副调节器的给定,而副调节器的输出则作用到执行机构去控制生产过程,因此称之为串级控制。并且副回路还可以根据情况选择多个,形成多串级控制系统。

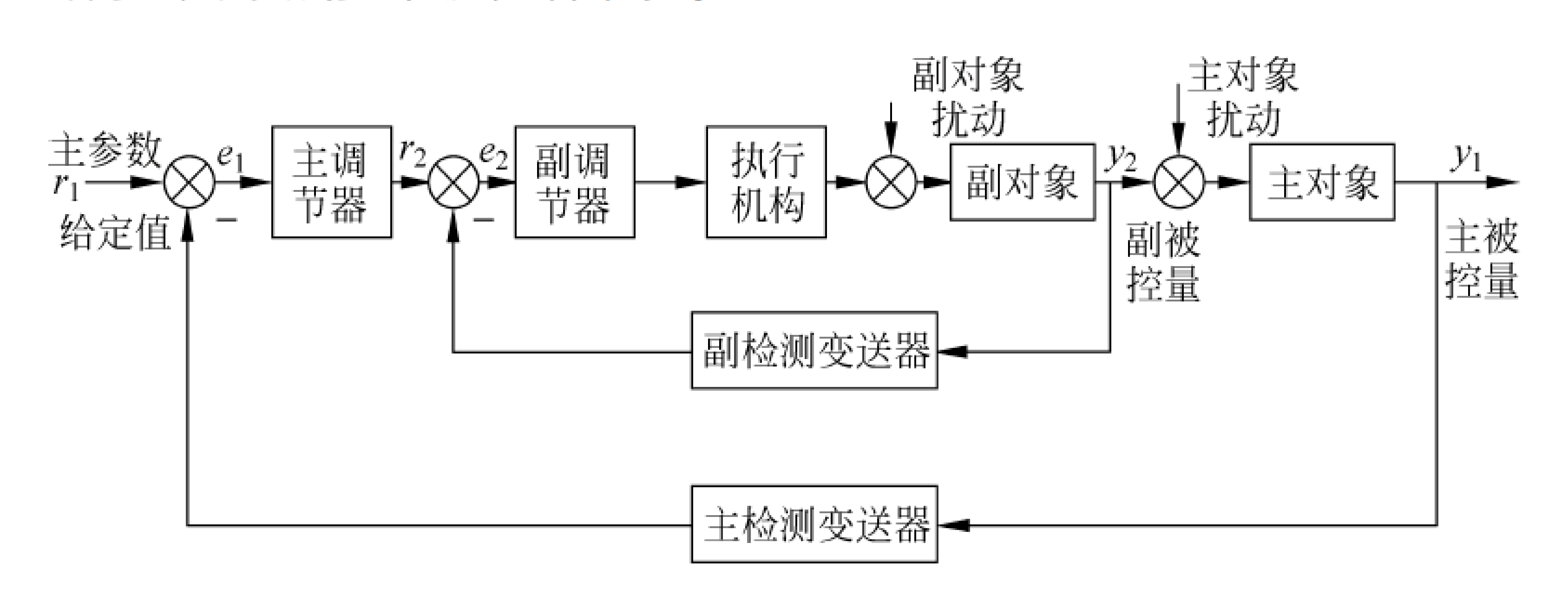


图 5.24 一般串级控制系统结构框图

整个闭环副回路可以作为一个等效的对象来处理,因此主回路的设计与一般的单回路控制系统没有太大的区别,无需特殊讨论。但是副回路的设计和参数选择是系统设计和实施中都应予以考虑的问题。串级控制比单回路控制有更强的抑制扰动的能力,通常副回路抑制扰动的能力比单回路控制能高出几十倍乃至上百倍,这些串级控制系统的一系列优点都是因为增加了副回路的缘故,因此副回路设计的质量是保证发挥串级系统优点的关键所在。在设计串级控制系统时应遵循以下的原则。

(1) 将系统中的主要扰动包含在副回路中。把主要扰动包含在副回路中,可在扰动影响主回路被控参数之前,通过副控制回路的调节作用,使扰动的影响大大削弱。

(2) 当主回路的积分控制造成的相角滞后对系统的影响较大时,应该尽量将积分环节包含于副回路中,相角滞后将可以减少,有利于改善调节系统的品质。

(3) 必须有一个可以测量的中间变量作为副回路被控参数,或选择可以通过观测分析由下游状态推断上游状态的中间变量。

(4) 主、副控制回路采样周期相同时称为同步采样,主、副回路采样周期不同时称为异步采样。主、副控制回路的采样周期不同时,应该 $T_{主} \geqslant 3T_{副}$,即二者之间相差 3 倍以上,以避免主控回路和副控回路之间发生相互干扰或共振。

在串级控制中,主、副调节器的选型很重要。由于主调节器和副调节器的任务不同,对它们调节规律的选取也有不同考虑。主调节器的任务是准确保持被控参数符合生产要求,因此为了减少稳态误差,提高控制精度,应具有积分控制,同时为了使系统反应灵敏,动作迅速,应加入微分控制,所以一般主调节器应采用 PID 控制规律。对于副调节器,其主要任务是要快速动作以迅速抵消落在副环内的扰动,而且副参数一般并不要求无差,所以副调节器一般采用 P 或 PI 控制规律,较少采用 PID 控制规律。

串级控制系统的参数整定要比单回路系统复杂些。因为两个调节器串在一起,在一个系统中工作,所以相互之间或多或少总有些联系和影响。同时串级系统应用也比较成熟,所以其参数整定的方法也很多,但不管用哪种方法,一般的原则都是先副回路再主回路,由内

层向外层逐层进行。

5.6.2 数字串级控制算法

根据图 5.24 可知，如果主调节器和副调节器都由计算机来实现时，则构成的计算机串级控制系统如图 5.25 所示。为了使问题简化，图中忽略了执行器和测量变送器。

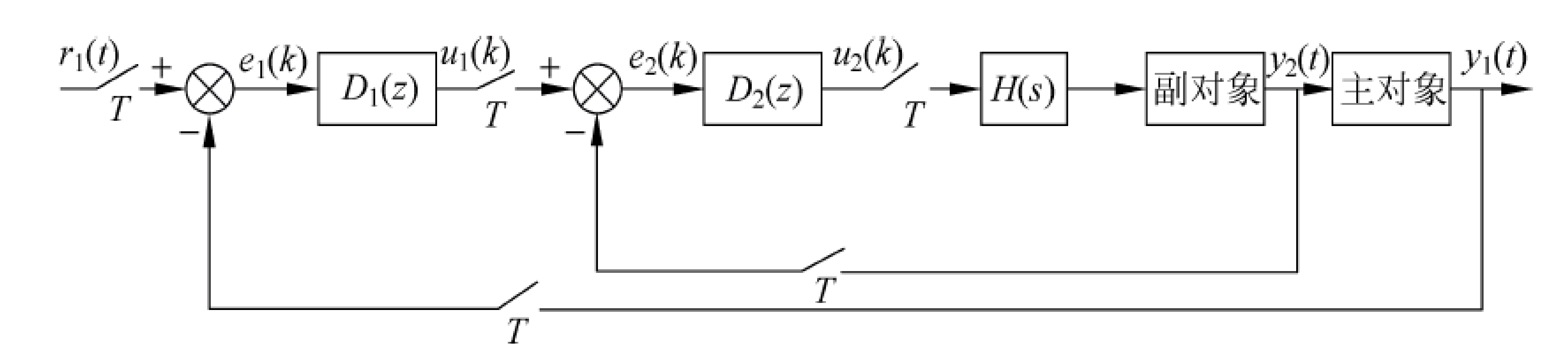


图 5.25 计算机串级控制系统结构图

串级控制系统的控制方式有两种：一种是同步采样控制，也即主、副控制回路的采样周期相同，但因为一般副对象的响应速度较快，故应以副回路为准。另一种是异步采样控制，即主回路的采样控制周期是副回路采样控制周期的整数倍。这是考虑到一般串级系统中主对象的响应速度慢，副对象的响应速度快的缘故。则对于图 5.25，设主、副控制回路采样周期相同，均为 T，$H(s)$为零阶保持器，$D_1(z)$和 $D_2(z)$控制器均采用数字 PID 控制规律。

在计算机控制系统中，不管串级控制有多少级，计算的顺序总是从最外面的回路向内回路进行。则对于图 5.25 所示的双回路串级控制系统在每个采样周期的计算顺序如下。

(1) 计算主回路的偏差 $e_1(k)$

$$e_1(k) = r_1(k) - y_1(k) \tag{5.107}$$

式中，$r_1(k)$为主回路给定值；$y_1(k)$为主回路的被控参数。

(2) 计算主回路 PID 控制器的输出 $u_1(k)$

$$u_1(k) = u_1(k-1) + \Delta u_1(k) \tag{5.108}$$

$$\begin{aligned}\Delta u_1(k) = & K_{P1}[e_1(k) - e_1(k-1)] + K_{I1}e_1(k) \\ & + K_{D1}[e_1(k) - 2e_1(k-1) + e_1(k-2)]\end{aligned} \tag{5.109}$$

式中，K_{P1}为比例增益；K_{I1}为积分系数，$K_{I1}=K_{P1}\dfrac{T}{T_{I1}}$；$K_{D1}$为微分系数，$K_{D1}=K_{P1}\dfrac{T_{D1}}{T}$。

(3) 计算副回路的偏差 $e_2(k)$

$$e_2(k) = u_1(k) - y_2(k) \tag{5.110}$$

(4) 计算副回路 PID 控制器的输出 $u_2(k)$

$$u_2(k) = u_2(k-1) + \Delta u_2(k) \tag{5.111}$$

$$\begin{aligned}\Delta u_2(k) = & K_{P2}[e_2(k) - e_2(k-1)] + K_{I2}e_2(k) \\ & + K_{D2}[e_2(k) - 2e_2(k-1) + e_2(k-2)]\end{aligned} \tag{5.112}$$

式中，K_{P2}为比例增益；K_{I2}为积分系数，$K_{I2}=K_{P2}\dfrac{T}{T_{I2}}$；$K_{D2}$为微分系数，$K_{D2}=K_{P2}\dfrac{T_{D2}}{T}$。

(5) 副回路控制器的输出 $u_2(k)$送到 D/A 转换器，经 D/A 转换成模拟信号输出驱动执行机构，去控制被控对象。当执行机构可接受数字信号时，则可不必进行 D/A 转换，直接将 $u_2(k)$送去控制执行机构。

5.6.3 副回路微分先行串级控制算法

为了防止主调节器输出(也就是副控制器的给定值)变化过大而引起副回路的不稳定,同时,也为了克服副对象惯性较大而引起调节品质的恶化,加快响应速度,改善控制质量。在副回路的反馈通道中加入微分控制环节,称为副回路微分先行,系统的结构如图5.26所示。

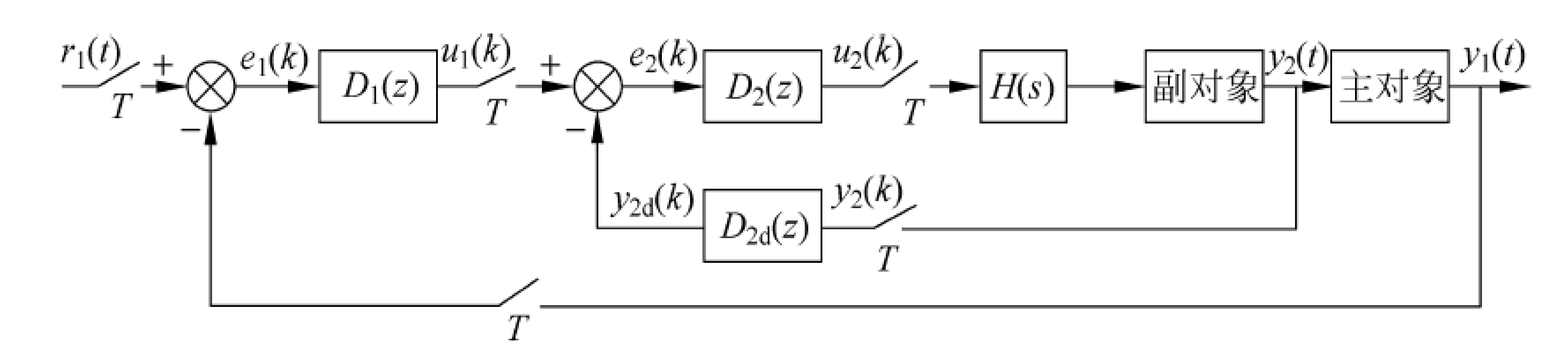


图5.26 副回路微分先行计算机串级控制系统

微分先行部分的传递函数为

$$D_{2d}(s)=\frac{Y_{2d}(s)}{Y_2(s)}=\frac{T_2s+1}{\alpha T_2s+1} \tag{5.113}$$

式(5.113)中,α 为微分放大系数。式(5.113)相应的微分方程为

$$\alpha T_2\frac{\mathrm{d}y_{2d}(t)}{\mathrm{d}t}+y_{2d}(t)=T_2\frac{\mathrm{d}y_2(t)}{\mathrm{d}t}+y_2(t) \tag{5.114}$$

将式(5.114)写出差分方程为

$$\alpha T_2\left[\frac{y_{2d}(k)-y_{2d}(k-1)}{T}\right]+y_{2d}(k)=T_2\left[\frac{y_2(k)-y_2(k-1)}{T}\right]+y_2(k) \tag{5.115}$$

整理可得

$$\begin{aligned}y_{2d}(k)&=\frac{\alpha T_2}{\alpha T_2+T}y_{2d}(k-1)+\frac{T_2+T}{\alpha T_2+T}y_2(k)-\frac{T_2}{\alpha T_2+T}y_2(k-1)\\&=\Phi_1y_{2d}(k-1)+\Phi_2y_2(k)-\Phi_3y_2(k-1)\end{aligned} \tag{5.116}$$

式(5.116)中,$\Phi_1=\frac{\alpha T_2}{\alpha T_2+T}$,$\Phi_2=\frac{T_2+T}{\alpha T_2+T}$,$\Phi_3=\frac{T_2}{\alpha T_2+T}$。

系数 Φ_1、Φ_2、Φ_3 可先进行离线计算,并存入到内存中指定的单元,以备控制计算时用。下面给出副回路微分先行的串级控制计算步骤(假设主控制器采用PID控制规律,副控制器采用PI控制规律)。

(1) 计算主回路的偏差 $e_1(k)$

$$e_1(k)=r_1(k)-y_1(k) \tag{5.117}$$

(2) 计算主回路PID控制器的输出 $u_1(k)$

$$u_1(k)=u_1(k-1)+\Delta u_1(k) \tag{5.118}$$

$$\begin{aligned}\Delta u_1(k)=&K_{P1}[e_1(k)-e_1(k-1)]+K_{I1}e_1(k)\\&+K_{D1}[e_1(k)-2e_1(k-1)+e_1(k-2)]\end{aligned} \tag{5.119}$$

(3) 计算副回路微分先行部分的输出 $y_{2d}(k)$

$$y_{2d}(k)=\Phi_1y_{2d}(k-1)+\Phi_2y_2(k)-\Phi_3y_2(k-1) \tag{5.120}$$

(4) 计算副回路的偏差 $e_2(k)$

$$e_2(k) = u_1(k) - y_{2d}(k) \tag{5.121}$$

(5) 计算副回路 PI 控制器的输出 $u_2(k)$

$$u_2(k) = u_2(k-1) + \Delta u_2(k) \tag{5.122}$$

$$\Delta u_2(k) = K_{P2}[e_2(k) - e_2(k-1)] + K_{I2}e_2(k) \tag{5.123}$$

(6) 副回路控制器的输出 $u_2(k)$经 D/A 转换成模拟信号输出驱动执行机构,去控制被控对象。当执行机构可接受数字信号时,则可不必进行 D/A 转换,直接将 $u_2(k)$送去控制执行机构。

除了前面介绍的常规以及复杂控制技术外,还有预测控制、模糊控制和专家控制等先进控制技术,读者可以查阅相关参考资料。

习题

1. 试求单位阶跃函数的 Z 变换。

2. 求 $f(t)=t^2\mathrm{e}^{-aT}$ 的 Z 变换。

3. 数字控制器的连续化设计步骤是什么?

4. 什么是数字 PID 的位置增量型算法?它们的优缺点各是什么?

5. 什么是积分饱和现象?它是如何引起的?可以采用什么办法消除?

6. 在数字 PID 中,采样周期 T 的选择需要考虑哪些因素?

7. 试分析基本 PID 算法与不完全微分 PID 算法的控制作用有何区别?不完全微分 PID 算法的优越性是什么?

8. 何为 PID 控制器参数的整定?整定方法有哪些?整定的实质是什么?

9. 数字控制器的离散化设计步骤是什么?

10. 什么是最少拍控制?最少拍控制的性能指标有哪些?

11. 最少拍系统有哪些局限性?对输入函数的适应性如何?

12. 试简述设计最少拍系统的约束条件有哪些?

13. 已知某被控对象的传递函数为 $G_c(s)=\dfrac{5}{s(0.1s+1)(0.05s+1)}$,采用零阶保持器,设采样周期 $T=0.1\mathrm{s}$。针对单位速度输入设计最少拍有纹波和无纹波数字控制器 $D(z)$,计算采样时刻数字控制器和系统的输出响应,并绘制图形。

14. 已知某计算机控制系统的被控对象传递函数为 $G_c(s)=\dfrac{\mathrm{e}^{-3s}}{5s+1}$,采用零阶保持器,设采样周期 $T=1\mathrm{s}$。针对单位阶跃输入函数,设计最少拍无纹波数字控制器 $D(z)$,计算采样瞬间数字控制器和系统的输出响应,并绘制图形。

15. 已知某被控对象的传递函数为 $G_c(s)=\dfrac{1}{s^2}$,采样周期 $T=1\mathrm{s}$,采用零阶保持器,针对单位速度输入函数,按以下要求设计。

(1) 用最少拍无纹波系统的设计方法,设计 $\Phi(z)$ 和 $D(z)$。$D(z)=\frac{1}{G(z)}\frac{\Phi(z)}{1-\Phi(z)}$。

(2) 求出数字控制器输出序列 $u_2(k)$ 的递推形式。

(3) 画出采样时刻数字控制器和系统的输出曲线。

16. 简述史密斯预估补偿器的基本思想。

17. 试画出计算机串级控制系统的方框图,并简述其原理。

18. 简述串级控制系统在每个采样周期的计算步骤。

第6章 MATLAB与控制系统仿真

MATLAB是"矩阵实验室"的缩写,它的基本运算单元为矩阵,是一种功能非常强大的科学计算与仿真软件,也是目前国际上控制领域首选的计算机语言。本章主要介绍MATLAB语言的环境、常用语句及其用法、基本运算、基本编程方法、绘图方面的应用,以及控制系统计算机建模与仿真的基本知识。

6.1 MATLAB软件简介

6.1.1 MATLAB的发展历程及特色

MATLAB是由美国MathWorks公司发布的主要面对科学计算、可视化以及交互式程序设计的软件,最早由美国的Clever Moler博士于1980年开发,设计者的初衷是为解决"线性代数"课程的矩阵运算问题,取Matrix和Laboratory两个词的前3个字母组合而成,为"矩阵实验室"的意思。MathWorks公司在1992年推出了具有划时代意义的、告别DOS版的MATLAB 4.0版本,随后又紧接着推出了5.x、6.x以及目前的7.x版本。

MATLAB作为目前国际上最流行的科学计算与程序设计软件,它集计算机语言和设计、仿真、测试、实现为一体,在科学运算、数据处理、图像处理、自动控制、系统仿真等领域得到了广泛的应用。

MATLAB的特色优势如下。

(1) 高效的数值及符号计算功能,使用户从繁杂的数学运算分析中解脱出来。

(2) 具有完备的图形处理功能,实现计算结果和编程的可视化。

(3) 友好的用户界面及接近数学表达式的自然化语言,使学者易于学习和掌握。

(4) 可视化建模和强大的动态系统仿真功能。

(5) 庞大的应用工具箱,为用户提供了大量方便实用的处理工具。

6.1.2 MATLAB的桌面

当成功启动MATLAB软件后,将进入MATLAB的集成开发环境,也就是用户面对的MATLAB的桌面,默认的MATLAB桌面结构如图6.1所示。

MATLAB的集成开发环境主要包括主窗口(Main Window)、命令窗口(The Command Window)、命令历史窗口(The Command History Window)、启动平台(Launch Pad)、编辑

调试窗口(The Edit/Debug Window)、工作空间浏览器和矩阵编辑器(Workspace Browser and Array Editor)、帮助空间窗口(Help Browser)、当前路径窗口(Current Directory Browser)等。下面详细介绍几个主要的常用窗口。

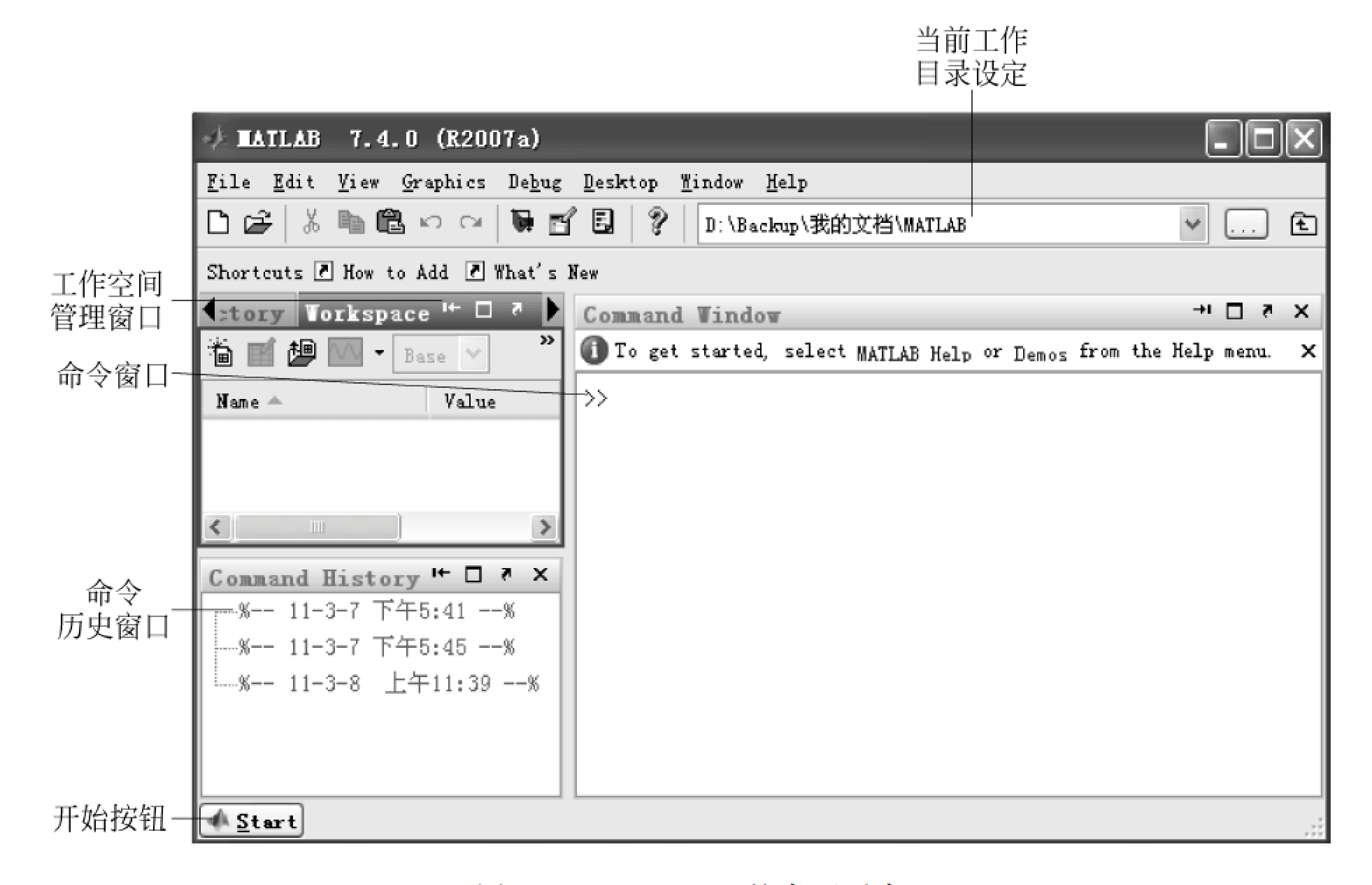


图 6.1 MATLAB 的桌面平台

1. MATLAB 主窗口

MATLAB 主窗口是 MATLAB 的主要工作界面,它集成管理各个嵌入的子窗口,不能进行任何的计算机任务的操作,只进行一些工作环境参数的设置。主要包括菜单栏和常用的工具栏。

2. 命令窗口

MATLAB 桌面的右边是命令窗口。命令窗口是 MATLAB 最主要的命令交互和工作窗口,用来输入用户命令并可以显示除图形以外的所有的执行结果。命令窗口中的“>>”为运算提示符,用户在它后面输入一系列的命令并按 Enter 键后,MATLAB 就会解释执行所给的命令,并且在这个窗口中给出计算结果,然后再次进入命令提示符状态,等待下次的输入。

一般来说,命令执行的结果可以在屏幕上显示,也可以不显示,主要由逗号和分号来区分,如果指令后面以分号结束,屏幕上将不显示结果。同时逗号和分号还可以作为指令间的分隔符,以使多条语句在同一行出现。

MATLAB 命令窗口的一个命令行可以输入一条命令,也可以输入若干条命令,如果一个命令行比较长,一个实际的物理行之内写不下,可以在第 1 个物理行之后加上 3 个小黑点并按 Enter 键,然后接着下一个物理行继续写命令的其他部分,系统会自动地认为整个是一个逻辑行命令。3 个小黑点称为续行符,即把下面的物理行看作该行的逻辑继续。

在 MATLAB 里，还有许多的方向键和控制键用来实现对命令行的快速编辑。

3. 历史窗口

在默认设置下，历史记录窗口中会自动保留自安装起所有用过的命令的历史记录，并标明使用的时间，这极大地方便了使用者查询。而且，通过双击某一行命令，即可进行历史命令再运行，无须在命令窗口中再输入。如果要清除这些历史记录，可以在主窗口的 Edit 菜单下选择 Clear Command History 菜单即可。

4. 工作空间窗口

工作空间窗口是 MATLAB 的重要组成部分，用于存储显示目前内存中各种变量的名称、大小、字节数及变量类型说明和结果。在该窗口中可对变量进行观察、编辑、保存和删除。

启动 MATLAB 软件后，会自动地建立一个工作空间来存储各种变量和结果，并且在 MATLAB 运行期间一直存在，关闭 MATLAB 后工作空间会自动消失。

5. 当前目录窗口

当前目录是指 MATLAB 运行文件时的工作目录，在当前目录窗口中可以显示或改变当前目录，还可以显示当前目录下的文件并提供搜索功能。

MATLAB 运行时，只有存放在当前目录或搜索路径下的函数、文件才可以被调用或运行，否则就会提示出现错误。

如果用户需要修改当前工作目录，可以利用 MATLAB 桌面上的当前工作目录设定区进行修改设置，也可以在命令窗口通过 cd 命令来设置。例如，将用户目录 d:\liu\MATLAB 设置为当前目录，可在命令窗口输入以下命令：

```
>> cdd:\liu\MATLAB
```

6. 启动 Start 按钮

MATLAB 主窗口左下角还有一个 Start 按钮，单击该按钮会弹出一个菜单，选择其中的命令即可快速执行 MATLAB 的各种工具，并且可以查阅 MATLAB 包含的各种资源。

7. 帮助系统

MATLAB 中包含有大量的工具箱和函数，而且随着软件版本的不断升级，这些工具箱和函数也在不断地扩充和丰富。用户在学习 MATLAB 的过程中，想掌握每一个函数和工具箱是不太可能的，因此就要经常借助 MATLAB 软件自身提供的帮助系统来学习这些函数和工具箱的使用方法，理解、掌握和熟练运用 MATLAB 的帮助系统是非常重要的。

帮助可以从命令窗口获得，也可以从主窗口的菜单中获得。在命令窗口获得帮助比较快捷，直接在命令窗口输入“help”或“help＋需要帮助的命令”即可显示相关内容的帮助。在主窗口的 Help 菜单中获得帮助，要通过菜单去一级级地寻找选择想要帮助的内容。

6.1.3　MATLAB 的基本运算

启动 MATLAB 软件之后，在 MATLAB 的桌面右边会看到一个 Command Window 的命令窗口，它是最主要的窗口，既是输入命令也是显示计算结果的地方，在这个地方输入命令并执行，显示直观形象，但相对来说只适合于比较短的命令程序。另外，MATLAB 还专门有一个编程窗口，也即 M-file 的程序书写方式，可以自行将命令或者算式写成程序，然后存储成 M 文件，用来编辑比较长的应用程序。

MATLAB 有 3 种进行计算的方法，第 1 种就如同使用计算器一样，直接输入所需要的数值和运算符，即可立即从屏幕上获得结果。第 2 种先对变量赋值，然后再输入由变量和运算符构成的表达式，也可立即获得结果。第 3 种，采用如同高级语言编程的方法在 M-file 中来编写程序，用来解决含有循环、迭代、判断、递归等算法比较复杂的问题。上述 3 种方法中，第 2 种和第 3 种还包括了矩阵和数组运算，只要定义了数组和矩阵变量，就可以如同普通代数运算一样直接用变量进行数学运算，十分方便。

1. 直接运算

在 MATLAB 下进行基本数学运算，只需在命令窗口提示号“＞＞”之后，将运算式和运算符直接输入，并按 Enter 键即可，MATLAB 会将计算的结果以 ans 默认的结果变量存储表示，并将 MATLAB 运算后的答案(Answer)显示在屏幕上。

例如：

```
>>(9 * 7 + 5 - 3)/5
ans  = 13
```

MATLAB 提供了丰富的运算符，使它的使用更加方便简单。

(1) 算术运算符如表 6.1 所示。

表 6.1　MATLAB 基本算术运算符

符　号	功　能	符　号	功　能
+	加法	−	减法
*	乘法	/	右除
\	左除	^	幂运算

(2) 关系运算符如表 6.2 所示。

表 6.2　MATLAB 关系运算符

符　号	功　能	符　号	功　能
＜	小于	＜＝	小于等于
＞	大于	＞＝	大于等于
＝＝	相等	～＝	不等于

注意：关系相等运算符“= =”和赋值运算符“=”在运算中不要搞混淆。

(3) 逻辑运算符如表6.3所示。

表6.3 MATLAB逻辑运算符

符号	功能	符号	功能
&	逻辑与运算	~	逻辑非运算
|	逻辑或运算	xor	逻辑异或运算

2. 变量运算

在MATLAB中使用变量之前，不需要对变量指定数据类型，也不必事先进行变量声明。但在MATLAB中对变量名称还有一些规定。

(1) 变量名称的英文字母区别大小写，比如ans、Ans、anS是不同的3个变量。

(2) 变量名的长度上限为31个字符，第31个字符之后的字符将被忽略。

(3) 变量名的第1个字母必须是英文，随后可以掺杂英文字母、数字或下划线，但是不能使用标点符号。

同时，由于MATLAB中有一些系统默认的预定义的变量和系统函数，在编写命令和程序时，也应尽量避免使用预定义的变量和函数名，以免混淆，如果变量名与MATLAB中的函数或命令重名，那么这个函数或命令将不能被访问。表6.4给出了MATLAB语言中预定义的变量及其说明。

表6.4 MATLAB预定义变量

符号	功能	符号	功能
i,j	虚数单位	Inf	无穷大的数∞
pi	圆周率π值	NaN	Not-a-Number不定值
ans	计算结果的默认变量名	eps	浮点数相对精度

例如在命令窗口输入pi，则显示如下：

```
>> pi
ans = 3.1416
```

MATLAB中变量的运算必须先赋值再运算，格式为：

```
变量名 = 运算表达式
```

比如：

```
>> a = 4;b = 5;c = a + b
c = 9
```

3. M文件运算

如果要计算的算式很长有数十行，或是需要一再执行的算式，前面在命令窗口输入的方式就显得不方便了。MATLAB提供了所谓的M-file的方式，可让使用者自行将指令及算式写成程序，然后储存成一个特别的文档，其扩展名是m，比如example.m，其中的example就是文件名称。同时文件命名时注意不要创建和MATLAB内建函数或命令同名的M文

件。当M文件保存完后，只需在命令窗口（The Command Windows）中输入这个M文件的名字，它就可以被执行了。

M文件可以分为M脚本文件(M_Scripts)和M函数(M_Functions)。M脚本文件也即M命令集，它的作用和将命令逐一在命令窗口输入是完全一样的。M函数可以根据程序的需要定义，但同时需要用到输入和输出参数来传递调用信息，如同其他高级语言中的函数一样。

1）编写M脚本文件的步骤

单击MATLAB指令窗口工具条上的New File图标，或者在MATLAB主菜单File下选择New，再选中M-file，就可打开如图6.2所示的MATLAB文件编辑器窗口，用户即可在空白窗口中输入程序代码，编写应用程序。

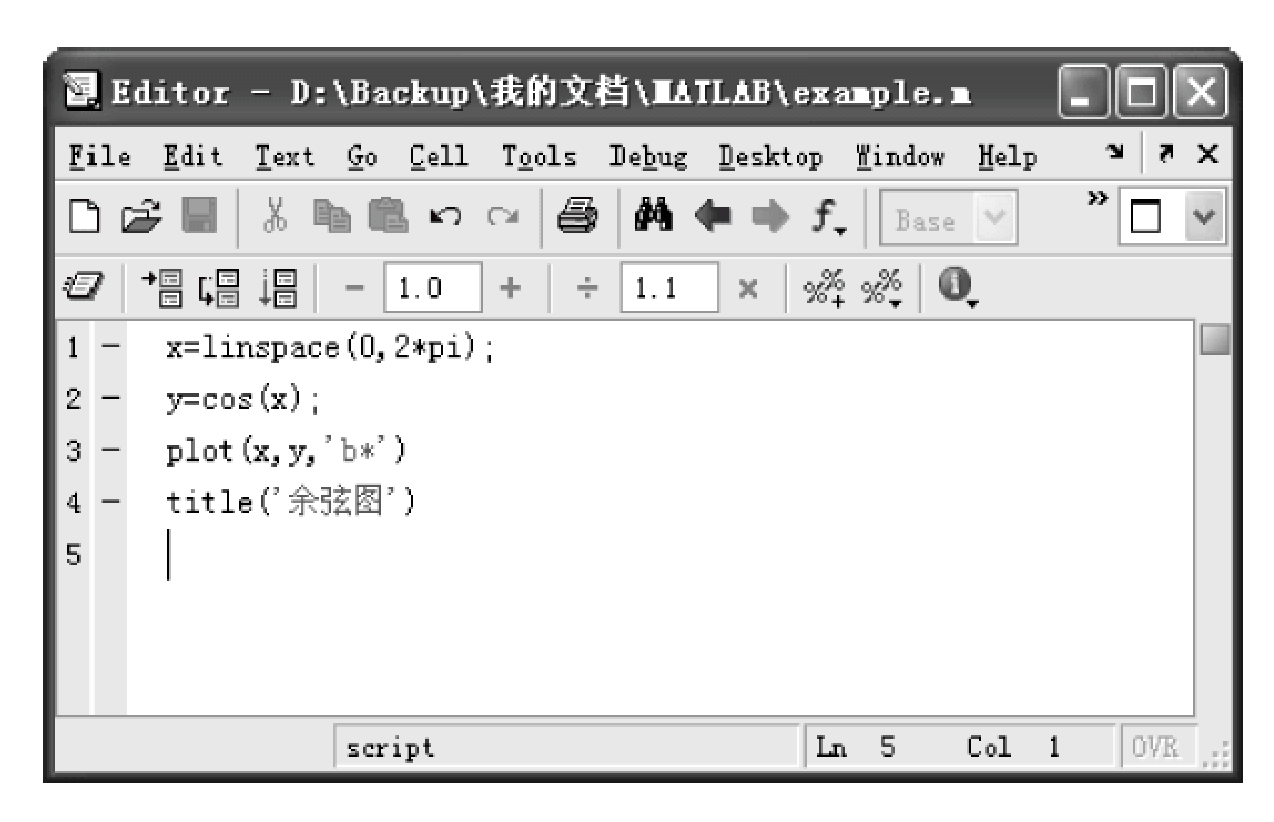


图6.2　MATLAB文件编辑器

输入如下一段程序：

```
x = linspace(0,2 * pi);
y = cos(x);
plot(x,y,'b * ');
title('余弦图');
```

编写完毕后单击File菜单下的Save命令，出现保存文件对话框，输入新编的文件名（如example），单击“保存”按钮，就完成了文件的保存。运行程序时，先使example.m所在的目录成为当前工作目录，或者让该目录处在MATLAB的搜索路径上，然后在命令窗口输入example文件名即可运行得到图形结果，如图6.3和图6.4所示。

2）编写M函数文件的步骤

M文件除了可以编写程序外，另外一个最重要的用途就是可以用来定义函数。函数是MATLAB语言中非常重要的组成部分，提供在各种工具箱中的M文件几乎都是以函数形式给出的，MATLAB的主体和各种工具箱本身就是极其庞大的函数库，当然，用户也可以按照函数格式要求编写定义各种函数，这样的函数可以和那些内建的函数一样方便使用。

MATLAB函数的一般格式为：

```
function [返回变量列表] = 函数名(输入参数表)
% 函数注释说明部分
函数体语句
end
```

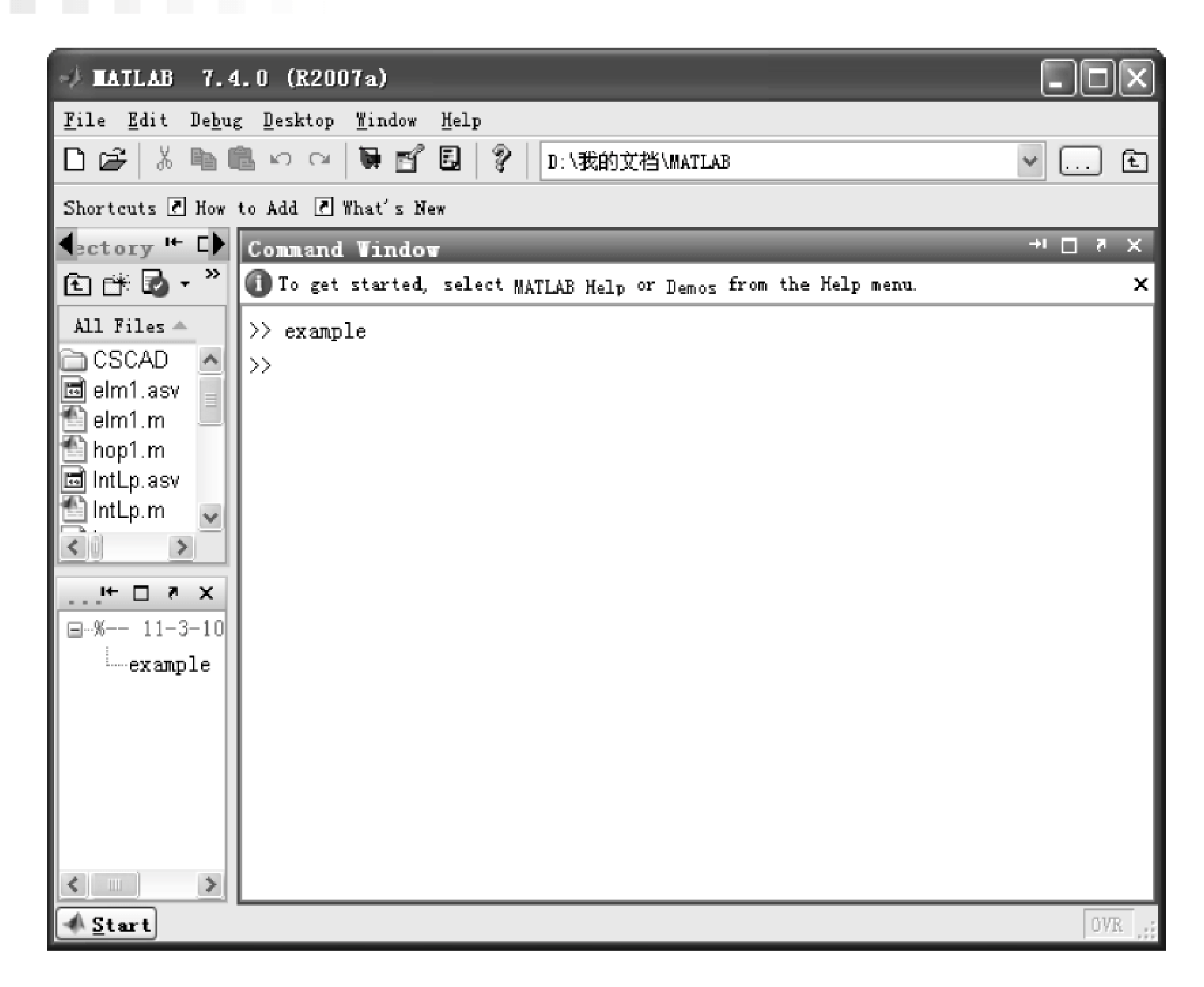


图 6.3 MATLAB 运行程序

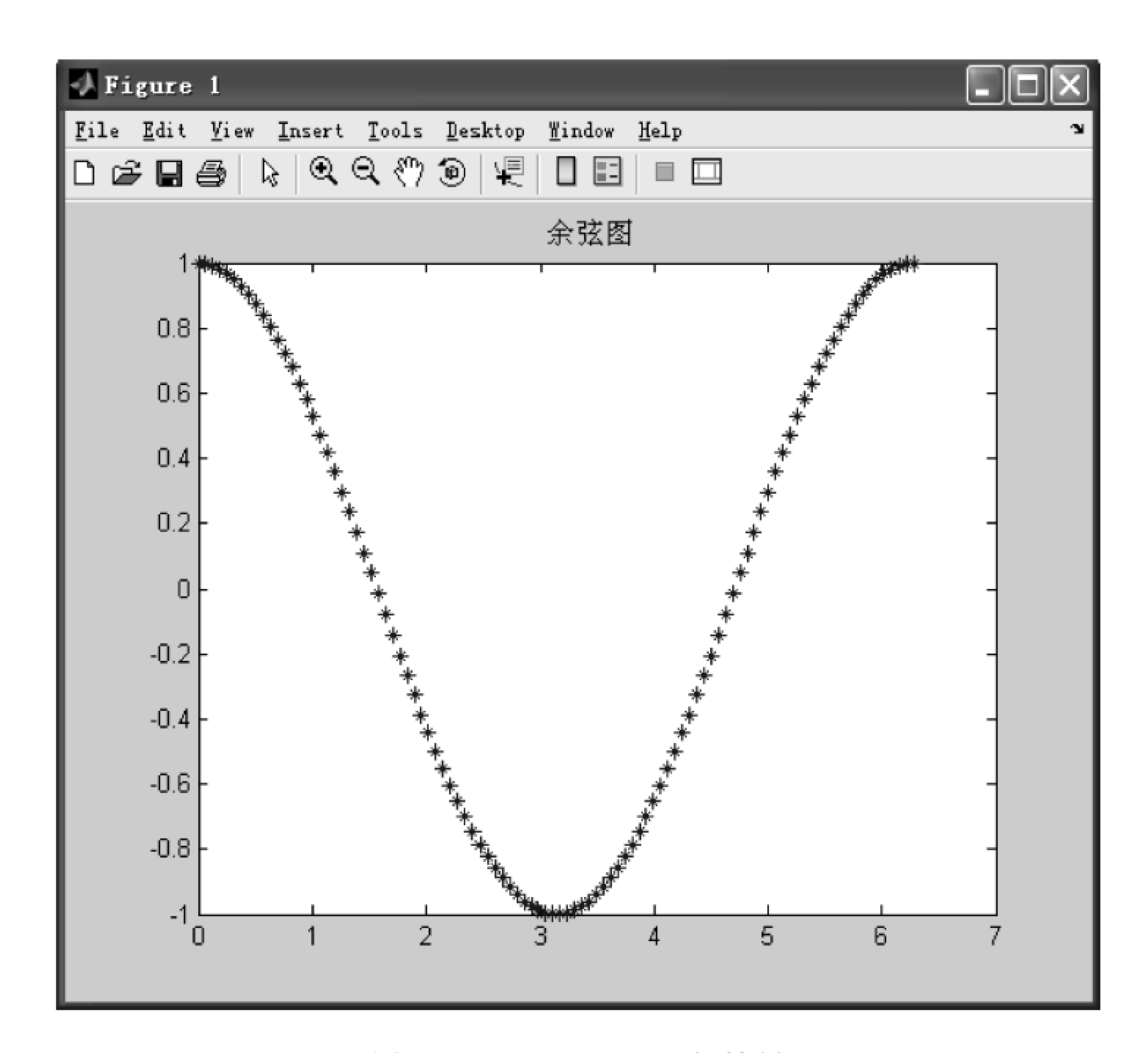


图 6.4 MATLAB 运行结果

M 函数定义的语法也有下面的一些规定，在使用时要加以注意。

(1) M 函数的第 1 行开始必须以 function 关键字开始，表示下面要建立的是函数文件。输入和输出表中如果有多个参数的话，以逗号分隔开。函数可以有零个或多个输入参量，也

可以有零个或多个输出参量。

(2) 函数名称的取名规定和一般变量命名相同。建立的函数名和保存的文件名必须相同。例如，函数 jisuan 必须存储为名在 jisuan.m 的文件。

(3) 函数体是函数的主体部分，函数体包括运算和赋值操作的所有程序，可以有程序流程控制、输入、输出、赋值、注释、计算等语句，还可以包括对其他函数的调用。

(4) 函数被调用时可以按照少于函数 M 文件中所规定的输入和输出的变量个数进行调用，但不能用多于函数 M 文件中所规定的输入和输出变量数目调用。如果调用时输入和输出变量数目多于在函数 M 文件中 function 语句所规定的数目，则调用后会自动返回一个错误。

(5) MATLAB 中把从百分号“%”开始到该行结束的部分作为注释，若注释有多行的话，则每行开头都要有“%”。注释不进行编译，仅作参考使用。

(6) 在函数文件中，除了函数定义行和函数体之外，其他部分都可以省略。

例如，计算 n 的阶乘函数 jc.m。

首先建立一个 M 函数：jc.m

```
function output = jc(n)
if n = = 1
output = 1; return;
end
output = n * jc(n-1);
end
```

建立的函数可在其他的 M 文件或者命令窗口调用。比如在命令窗口输入 jc(3)，就可以在该窗口立刻看到：

```
ans = 6
```

6.1.4　程序流程控制

作为一种交互式的程序设计语言，MATLAB 还提供了循环控制、条件转移、开关语句等，它们的语法格式与其他高级语言相似。

1. 循环结构

MATLAB 提供了两种循环结构，可以由 for…end 和 while…end 来实现，在这两个语句之间的部分称为循环体。for 语句使用灵活，常用于循环次数预先确定的情况，其使用格式为：

```
for 循环次数变量名 = 表达式
循环体语句
end
```

while 循环一般用于循环次数不能预先确定的情况，使用一个逻辑判断语句来决定循环的进行，其使用的基本形式为：

```
while 条件表达式
循环体语句
end
```

在 MATLAB 程序设计中，循环结构的执行速度较慢，所以在实际编写程序时，应尽量避开使用循环结构来实现，以提高程序执行的效率。

2. 条件语句结构

条件转移结构是一般程序设计语言都支持的结构，MATLAB 提供 if…end 结构来实现条件语句，同时也可以配合 else 语句和 else…if 语句来扩展结构。其使用格式为：

```
if(条件 1)
(条件 1 判断为 true 时执行的语句组)
else
(条件 1 判断为 false 时执行的语句组)
end
```

3. 开关语句结构

MATLAB 从 5.0 版本开始提供了开关语句结构，其基本格式为：

```
switch   开关表达式
case     表达式 1
语句段 1
case     表达式 2
语句段 2
⋮
otherwise
语句段 n
end
```

与其他的程序设计语言的 switch-case 语句不同的是，在 MATLAB 语言中，当其中一个 case 语句后的条件为真时，也即开关表达式的值等于 case 语句后的条件时，程序将转移到该语句去执行，执行完后直接跳出开关体，不再对其后的 case 语句进行判断，而执行 end 之后的语句。也就是说在 MATLAB 语言中，即使有多条 case 判断语句为真，也只执行所遇到的第 1 条为真的语句。这样就不必像 C 语言那样，在每条 case 语句后加上 break 语句以防止继续执行后面为真的 case 条件语句。

6.1.5 MATLAB 的绘图

MATLAB 不仅在数值计算方面功能强大，而且具有很强的图像处理显示功能，能以二维、三维乃至多维的形式显示图形。

1. 二维绘图

二维图形的绘制是 MATLAB 语言图形处理的基础，也是绘制复杂图形的基础。MATLAB 提供了多个函数用于图形的绘制。MATLAB 提供的最常用的绘制二维图形的命令是 plot 函数，其使用格式为：

```
plot(x,y,option)和 plot(x1,y1,option1, x2,y2,option2,…)
```

其中 x 和 y 均为事先定义好的要绘制的曲线上每一点的 x 和 y 坐标值，option 选项可以定义所绘制图形的颜色、线型和坐标点的标示符号，它必须用一对单引号括起来控制所显示图形的外观。上面两种格式可以绘制一条曲线和绘制多条曲线，并让每条曲线有不同的外观。上面是 plot 命令的完全格式，在实际应用中可以根据需要进行简化。比如：

```
plot(x,y); plot(x1,y1,x2,y2)
```

【例 6.1】 在 x 的区间 0～8 的范围内绘制正弦曲线 $y=\sin(x)$。

其程序为：

```
x = linspace(0,8,40);        % 在区间[0,8]生成一组 40 个线性等距的数值点
y = sin(x);
plot(x,y)
```

所画图形如图 6.5 所示。

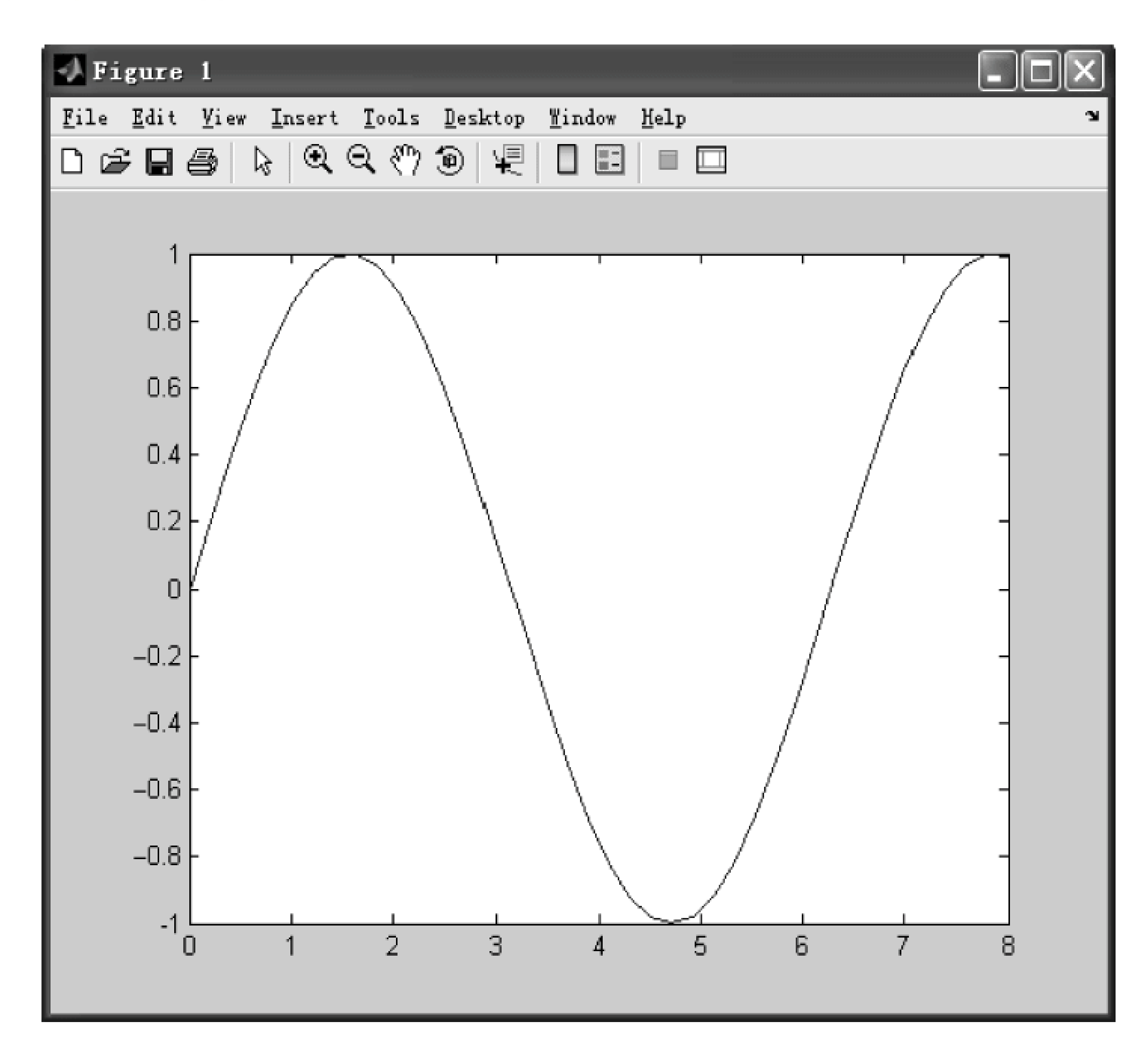


图 6.5 绘制正弦曲线

option 选项所定义的曲线的颜色、线型和标记点的表示符号可以任意组合，表 6.5 给出了其可以组合的选项。

表 6.5 线型、颜色、标记符号表

选 项	意 义	选 项	意 义
y	黄色	·	点号
m	品红	×	叉号
c	青色	o	圆圈
r	红色	+	加号
g	绿色	*	星号

续表

选　项	意　义	选　项	意　义
b	蓝色	s	方块
w	白色	d	菱形
k	黑色	v	上三角
—	实线	<	左三角
:	点线	>	右三角
—.	点划线	p	空心五角星
——	虚线	h	空心六角星
none	无线	^	下三角

比如：plot(x,y,'m * —. ')；就可以绘制出用“ * ”号标记的品红色的“点划”线。另外，MATLAB 在绘制完二维图形后，还给出了一些图形修饰与处理的命令。常用的命令有如下几项。

1）坐标网格显示

grid on：在当前图形中加网格线。

grid off：去掉当前图形中的网络线。

2）文字显示和标注

title('字符串')：当前坐标系顶部加标题。

xlabel('字符串')：当前坐标系 x 轴旁加标题(下方)。

ylabel('字符串')：当前坐标系 y 轴旁加标题(左侧)。

text(x,y,'字符串')：当前坐标系(x,y)指定位置处加文本。

gtext('字符串')：在鼠标单击处加文本，即文本交互式输入命令。

3）图例说明

legend('字符串 1','字符串 2',…,'字符串 n')：在当前图形内开启一个小视窗，建立一图例说明框。然后根据绘图命令的先后次序，用对应的字符串依次区分图形上的线，可用鼠标拖动图例框改变其位置。

legend off：删掉图例说明框。

4）设定坐标轴的范围

在默认状态下，绘图的坐标轴自动显示，自动刻度，并且以笛卡儿直角坐标系的形式显示。在 MATLAB 中用 axis([xmin,xmax,ymin,ymax])函数来调整图坐标轴的范围。

5）选择图形窗口

figure(n)：选择第 n 个图形窗口为当前窗口，若该窗口不存在，则创建它。

subplot(m,n,p)：将当前窗口划分为 $m\times n$ 个子坐标系统，并选择其中第 p 个坐标系统为当前坐标系统。各子坐标系统按行排序，编号分别为 1,2,…,$m\times n$。特别地 subplot(1,1,1)将删除所有子坐标系统而生成一个占满窗口的坐标系统。

hold on：保留当前图形及其坐标轴，允许后续图形附加到原图上。

hold off：其后的绘图命令将抹掉原图而重新绘制(默认设置)。

hold：on 与 off 的状态切换。

6）图形缩放

zoom on：使当前图形窗口的图形可缩放。在此状态下，单击放大图形，右击缩小图形，如果双击则恢复缩放前的大小。或者用鼠标拖动出一个矩形框而对需要的地方放大。

zoom off：关闭图形缩放功能状态。

zoom：在 on 与 off 状态之间进行功能切换。

2. 其他二维绘图命令

loglog(x,y)：x、y 轴均采用对数刻度绘制图形。

semilogx(x,y)：仅 x 轴采用对数刻度，y 轴仍采用线性刻度绘制图形。

semilogy(x,y)：仅 y 轴采用对数刻度，x 轴仍采用线性刻度绘制图形。

bar(x,y)：绘制长条状图。

stairs(x,y)：绘制阶梯状图。

stem(x,y)：绘制火柴棍图(离散信号常用此命令)。

polar(θ,ρ)：极坐标下的曲线绘制，在 θ 的范围内绘制 ρ 的曲线图。

pie(x)：绘制饼状图。

【例 6.2】 绘制如图 6.6 所示的正弦函数和余弦函数的分割子图，并在所绘的图形上加适当的标注。

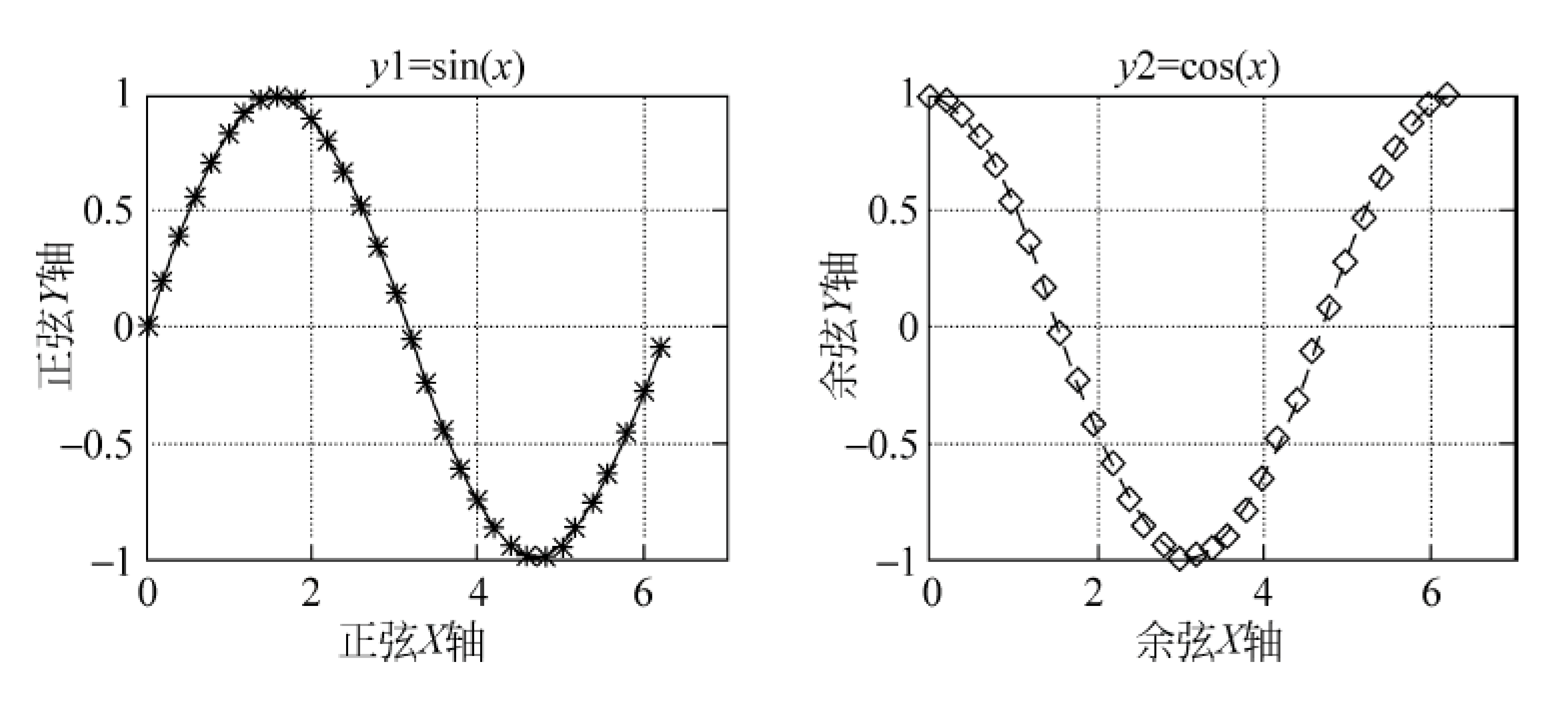


图 6.6 正弦函数和余弦函数分割子图

其 MATLAB 程序如下：

```
x = 0:0.2:2 * pi;           % 0 到 2 * pi 范围内每隔 0.2 取一个点
y1 = sin(x);                % 取样点处对应的正弦 y1 坐标
y2 = cos(x);                % 取样点处对应的余弦 y2 坐标
subplot(2,2,1);             % 一张画布分割成两个图形,并选中第 1 个子图作为当前图形
plot(x,y1,'k - * ');        % 绘制正弦函数,并用星号标记取样点
title('y1 = sin(x)');       % 给图形加标题
axis([0,7, - 1,1]);         % 限制坐标轴范围
xlabel('正弦 X 轴');        % x 轴坐标注解
ylabel('正弦 Y 轴');        % y 轴坐标注解
grid on;                    % 显示网格
subplot(2,2,2);             % 一张画布分割成两个图形,并选中第 2 个子图作为当前图形
plot(x,y2,'r -- d');        % 绘制余弦函数,并用菱形号标记取样点
title('y2 = cos(x)');       % 给图形加标题
axis([0,7, - 1,1]);         % 限制坐标轴范围
```

```
xlabel('余弦 X 轴');          % x 轴坐标注解
ylabel('余弦 Y 轴');          % y 轴坐标注解
grid on;                     % 显示网格线
```

3. 三维曲线绘图

MATLAB 除了在二维绘图方面具有强大的功能外，还提供了相应的三维绘图功能。MATLAB 语言中三维绘图和二维绘图在绘制方面有很多地方类似，而且图形属性的设置则完全相同。三维曲线绘图中经常用到的基本绘图命令为函数 plot3。

函数 plot3 是二维图形的绘制函数 plot 的特性扩展。函数格式除了增加第 3 维的信息之外，其余与二维函数 plot 相同。plot3 函数与 plot 函数用法十分相似，其调用格式为：

```
plot3(x1,y1,z1,选项 1,x2,y2,z2,选项 2,…,xn,yn,zn,选项 n)
```

其中每一组 x,y,z 组成一组三维曲线的坐标参数，选项的定义和 plot 函数完全相同，用来指定颜色、取样点、标记符号和线型，并且可以省略。当 x,y,z 是同维向量时，则 x,y,z 对应元素构成一条三维曲线。当 x,y,z 是同维矩阵时，则以 x,y,z 对应列元素绘制三维曲线，曲线条数等于矩阵的列数。

【例 6.3】 绘制三维空间螺旋线。

其 MATLAB 程序如下：

```
x = 0:pi/10:15 * pi;
y = sin(x);
z = cos(x);
plot3(x,y,z);
grid            % 添加网格
title('三维空间螺旋线');
xlabel('x');ylabel('sin(x)');zlabel('cos(x)');
```

程序所绘制的三维图形如图 6.7 所示。

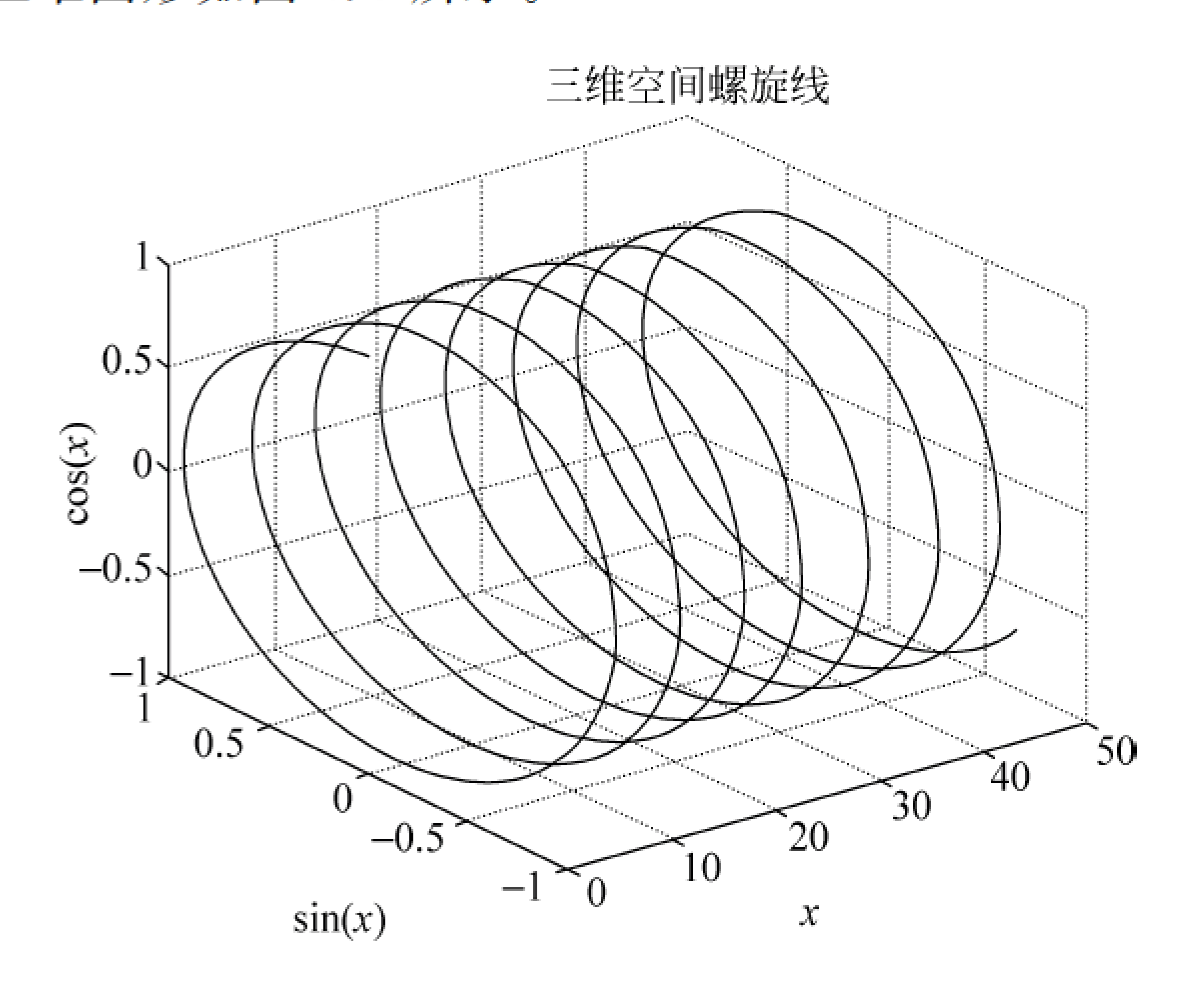


图 6.7　三维空间螺旋图

6.1.6 Simulink 简介

Simulink 是 MATLAB 中一个用来对动态系统进行建模、仿真和综合分析应用的附属软件包。它利用图形模块来形象地表示系统的各个环节，用箭头方向表示各个环节信号的输出及输入关系。Simulink 的出现，给控制系统的设计分析和仿真带来了极大的方便，它的名称也表明了该软件的两个主要功能，即 Simu（仿真）和 Link（连接）。Simulink 的模型表示方法与自动控制中常用的方块图表示法极为类似，因此可用鼠标在模型窗口上绘制出所需系统的模型，可以非常容易地将一个复杂的系统模型输入到计算机中，然后利用 Simulink 提供的功能来对系统进行仿真和分析。

Simulink 不但支持线性系统和非线性系统仿真，还可以进行连续系统和离散系统或者两者的混合系统仿真，同时它还支持具有多采样速率的系统仿真。也就是说，它可以用来模拟几乎所有可能遇到的动态系统。另外，Simulink 还提供了图形用户界面 GUI 以及图形动画的处理功能，能让用户非常方便地建立系统仿真模型并且直观地观察到仿真的整个过程。随着 MATLAB 软件版本的不断升级，Simulink 也在不断地扩大它的应用范围，它覆盖了通信、自动控制、视频和图形图像处理、电力系统等众多领域。

1. Simulink 模块库介绍

在 MATLAB 主界面的命令窗口下，输入 Simulink 回车后就可以打开 Simulink 模块库浏览器，或者单击 MATLAB 主界面工具栏上 Simulink 快捷按钮图标，也可同样打开模块库浏览器，模块库浏览器窗口列出了按功能分类的各种模块的名称，如图 6.8 所示。

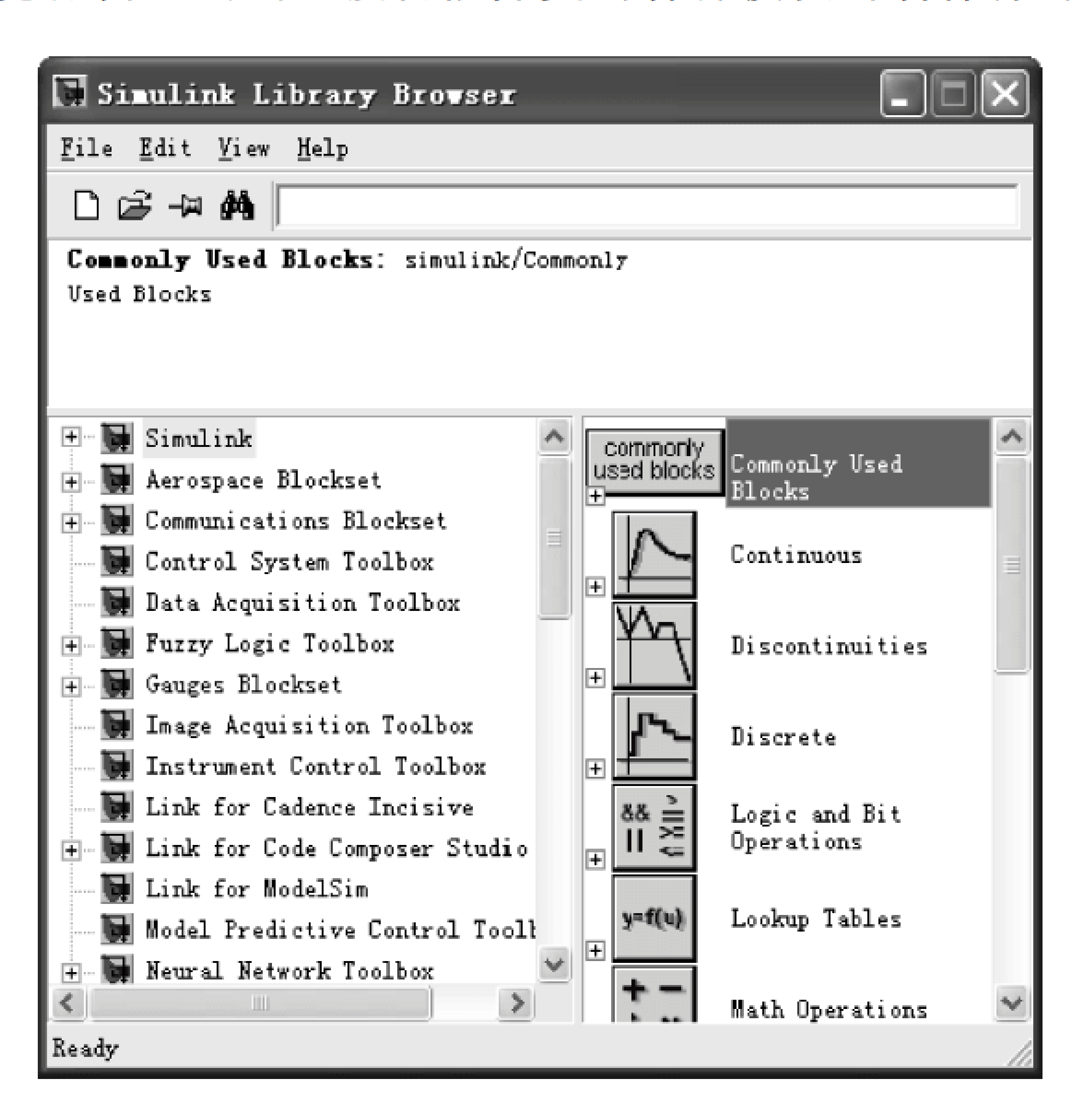


图 6.8 Simulink 模块库浏览器

在 Simulink 模块库浏览器中，双击 Simulink 可将 Simulink 库浏览树展开，可以看到在该模块库下按功能分类，包括 Continuous（连续系统模型库）、Discrete（离散系统模型库）、Function & Tables（函数与表模型库）、Math（数学运算模型库）、Nonlinear（非线性系统）、

Signal & System(信号与系统)、Sinks(输出节点库)和 Sources(输入源模块)等标准模块库,每个模块内又包含多个子模块,可以实现不同的功能连接。

2. Simulink 模块的操作处理

1) 标准模块的选取

搭建 Simulink 模型的许多操作,如复制模块或删除连线,都需要准确地选取这些对象。单击某一标准模块库,在 Simulink 模块库浏览器中会出现该模块库的所有模块,选取所需对象,单击鼠标即可选中,一直按下鼠标左键将该模块拖放至模型窗口中适当位置再松开鼠标左键,即完成模块的选取、复制。

2) 模块的编辑

模块的编辑包括模块的移动、复制、剪切和删除。所有的这些操作都与 Windows 环境下的操作一样。

(1) 模块移动:将光标置于要移动的模块并选中,按住鼠标左键不放将其拖曳到所需位置,松开左键即可完成移动。若要在移动中脱离连线,可先按住 shift 键,再进行拖曳即可。

(2) 模块复制:先用鼠标选中所要复制模块,然后按住鼠标右键进行拖曳移动即可复制一个一样的功能模块。

(3) 模块剪切:先选中模块,执行 Edit 下的 Cut 命令即完成对模块的剪切。

(4) 模块删除:先选中模块,执行 Edit 下的 Delete 命令即完成对模块的删除。

3) 模块外形的调整

(1) 改变模块方向:有时为了构建模型的方便,功能模块需要转向。在菜单 Format 下执行 Flip Block 命令,旋转模块 180°;执行菜单命令 Rotate Block,顺时针旋转模块 90°。

(2) 改变模块尺寸:该操作与在 Word 中改变图片尺寸的操作一样,先选中模块对象,对模块对象周围出现的 4 个黑色控制标记进行拖曳即可。

(3) 模块属性修改:对模块属性进行多种格式设置,均由菜单 Format 中对应的命令来完成。

(4) 改变模块的名称:模块下面都有相应的模块名称,可以根据实际情况对模块名称进行修改。用鼠标单击模块标题名称,使光标处于可编辑状态,Simulink 在模块名称外置一方框,此时可对模块名进行任意的修改,中英文皆可。最后在窗口的其他位置单击鼠标,修改工作完成。

4) 模块的连接

Simulink 模型的构建是通过用线将各种功能模块进行连接构成。将鼠标指针移到模块端口附近,当鼠标指针变为"+"时,按下左键不放拖曳鼠标到另一目的模块连接点,松开左键即可。同时双击连接线即可对连接线进行命名操作。

若要划分支线,则将鼠标指针移到要分支的线上,按下右键,拖动鼠标到另一连接点,松开右键即可。或者先按住 Ctrl 键,在要建立分支的地方按住鼠标左键拉出即可。

若要对线进行折弯,可以按住 Shift 键,再用鼠标左键在需要折弯的地方单击一下,就会出现一个小圆圈,表示此处是个折点,利用这个折点就可改变信号线的形状。

3. Simulink 模型的建立与运行

在 Simulink 中,简单模型的建立步骤如下:

（1）首先建立一个模型窗口，在此放置所需各模块。

（2）将所需功能模块由对应的模块库窗口复制到模型窗口。

（3）用连线连接模块，构成所需要的系统模型。

构建好一个系统的模型之后，就可以运行模型，得到仿真结果进行系统分析。运行一个系统模型仿真的完整过程步骤如下：

（1）设置仿真环境参数，主要在菜单 Simulation 下通过 Configuration Parameters 选项实现。

（2）启动仿真命令进行仿真，可以通过菜单 Simulation 下 Start 和在 MATLAB 命令窗口使用命令两种方式进行运行仿真。

（3）仿真结果分析。

下面通过一个例子来详细说明在 Simulink 中进行系统建模与仿真的过程。

【例 6.4】 建立如图 6.9 所示的 Simulink 系统仿真模型并进行单位阶跃响应仿真。

首先在 MATLAB 中启动 Simulink 模块库浏览器，然后在 File 菜单下选择 New，再选中 Model，建立一个新的模型文件，在空白的模型文件窗口建立仿真模型图，分别从相应的模型库中选择所需的元件和环节到 Model 中，并用线将各模块依次连接起来。所建立的模型如图 6.9 所示。

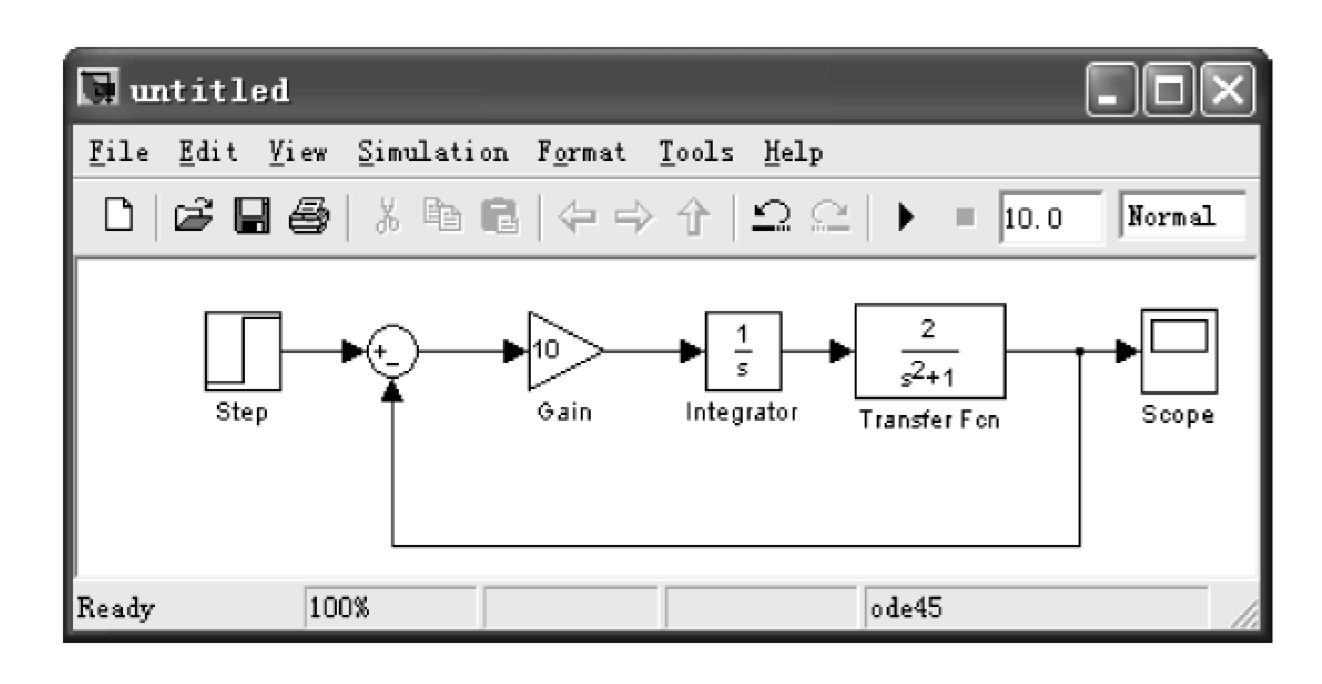


图 6.9　Simulink 模型图

单击 Simulation 菜单下的 Start 命令即可对系统模型按默认的仿真环境进行仿真，然后双击示波器 Scope 即可观察到系统的单位阶跃响应结果，如图 6.10 所示。

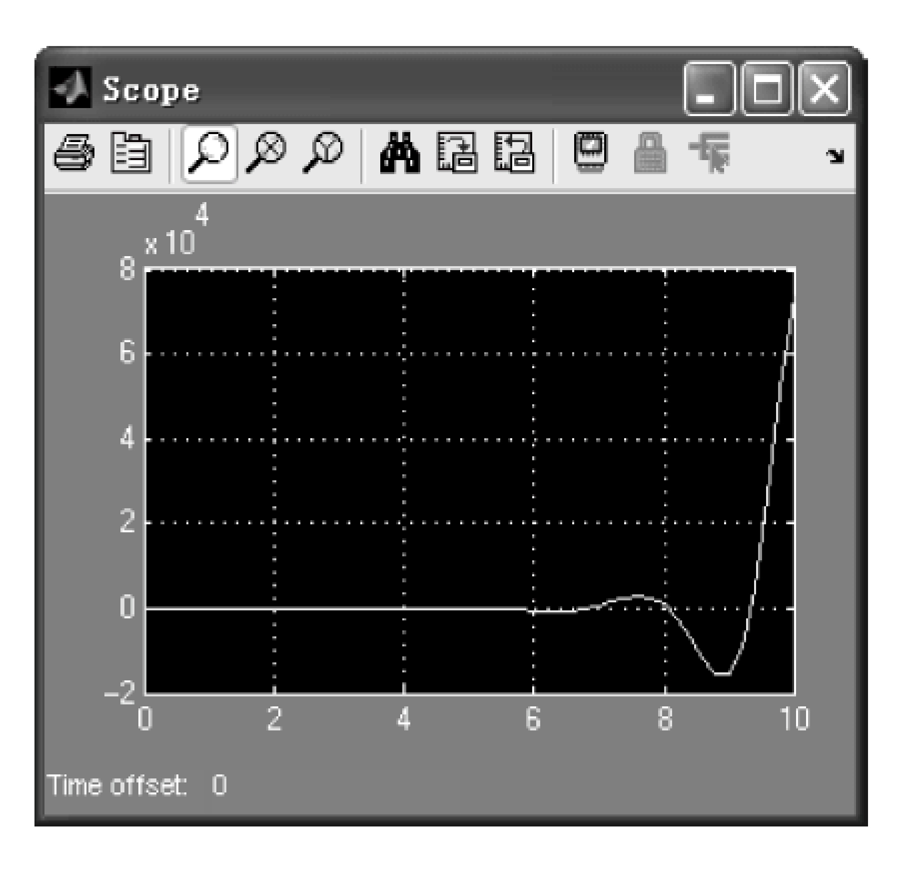


图 6.10　系统模型单位阶跃响应图

6.2 控制系统的建模及模型转换

控制系统的数学模型对于控制系统的研究有着重要的地位和作用。在控制系统的仿真分析研究中，首先要建立系统的数学模型，然后才能在此基础上建立系统的仿真模型，进行分析研究，并据此设计出合适的控制器对系统进行控制，使其达到预期的目标，这也符合工程实际的需要。

6.2.1 建立数学模型的方法

控制系统的数学模型是描述控制系统输入、输出以及内部物理量(或变量)之间关系的数学表达式。数学模型按系统运动特性分为静态数学模型和动态数学模型。在静态条件下(即变量各阶导数为零)，描述变量之间关系的代数方程叫静态数学模型；描述变量各阶导数之间关系的微分方程叫动态数学模型。

如果已知输入量及变量的初始条件，然后对微分方程求解就可以得到系统输出量的表达式，并由此输出表达式可对系统进行性能分析。因此，正确合理地建立控制系统的数学模型是分析和设计控制系统的首要工作。

控制系统数学模型的建立求取方法有分析法和实验法两种。

1. 分析法

分析法是对系统各部分的运动机理进行分析，根据它们所依据的物理规律或化学规律，确定系统中各变量之间的关系，分别列写相应的运动方程，从而建立数学模型。

2. 实验法

实验法是人为地给系统施加某种激励测试输入信号，测得它的输出响应，根据输入输出的数据结果，通过一定的数学处理方法用适当的数学模型去逼近，这种方法有时也称为系统辨识。

6.2.2 系统的时域模型

1. 连续系统

连续时间系统一般用微分方程来表示。对于单输入单输出系统，假设系统的输入信号为 $r(t)$，输出信号为 $y(t)$，其系统数学模型的微分方程可用下式表示为：

$$\begin{aligned}&a_n \frac{\mathrm{d}^n}{\mathrm{d}t^n}y(t)+a_{n-1}\frac{\mathrm{d}^{n-1}}{\mathrm{d}t^{n-1}}y(t)+\cdots+a_1\frac{\mathrm{d}}{\mathrm{d}t}y(t)+a_0 y(t)\\&=b_m\frac{\mathrm{d}^m}{\mathrm{d}t^m}r(t)+b_{m-1}\frac{\mathrm{d}^{m-1}}{\mathrm{d}t^{m-1}}r(t)+\cdots+b_1\frac{\mathrm{d}}{\mathrm{d}t}r(t)+b_0 r(t)\end{aligned} \tag{6.1}$$

式中，$a_i(i=0,1,\cdots,n)$ 和 $b_i(i=0,1,\cdots,m)$ 分别为输出和输入各阶导数项的系数。如果它们均为常数，就是线性定常微分方程，简称常微分方程。

2. 离散系统

线性离散时间系统常用差分方程来描述。对于单输入单输出系统，系统模型的一般形式为：

$$a_n y[(k+n)T]+a_{n-1}y[(k+n-1)T]+\cdots+a_1 y[(k+1)T]+a_0 y(kT)$$
$$=b_m r[(k+m)T]+b_{m-1}r[(k+m-1)T]+\cdots+b_1 r[(k+1)T]+b_0 r(kT) \qquad (6.2)$$

式中，y 是系统的输出；r 是系统的输入；a_n是输出系数；b_m 是输入系数。

同样，如果输入和输出的各项系数为常数，则它所描述的系统称为线性时不变系统(LTI)。MATLAB 控制工具箱提供了大量的工具函数方便对线性时不变系统进行建模分析和设计。

系统模型表示的微分方程和差分方程是描述系统动态特性的基本形式，经过不同的变换可得到系统数学模型的其他形式：传递函数、状态空间模型和零极点模型等。

6.2.3 系统传递函数模型

在初始条件为零时，线性系统输出量的拉普拉斯变换与输入量的拉普拉斯变换之比称为线性系统(或元件)的传递函数。

传递函数是经典控制理论中描述系统数学模型的一种表达形式，它表示了系统把输入量变换成输出量的传递关系。它只和系统本身结构、特性和参数有关，而与输入信号的形式无关。传递函数是研究线性定常系统动态响应和性能的重要工具。

1. 连续系统传递函数

对于连续系统，对公式(6.1)等号两边取拉普拉斯变换，并假设 y 与 r 的各阶导数的初始值均为零，则

$$a_n s^n Y(s)+a_{n-1}s^{n-1}Y(s)+\cdots+a_1 sY(s)+a_0 Y(s)$$
$$=b_m s^m R(s)+b_{m-1}s^{m-1}R(s)+\cdots+b_1 sR(s)+b_0 R(s) \qquad (6.3)$$

式中，$Y(s)$为输出 $y(t)$的拉普拉斯变换；$R(s)$为输入 $r(t)$的拉普拉斯变换。

从公式(6.3)可知其所描述的系统的传递函数为：

$$G(s)=\frac{Y(s)}{R(s)}=\frac{b_m s^m+b_{m-1}s^{m-1}+\cdots+b_1 s+b_0}{a_n s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0} \qquad (6.4)$$

在 MATLAB 语言中，可以利用分别定义的传递函数分子、分母多项式系数向量方便地对其加以描述。例如对于公式(6.4)，可以先分别定义传递函数的分子、分母多项式系数向量为：

$$\text{num}=[b_m, b_{m-1}, \cdots, b_1, b_0]$$
$$\text{den}=[a_n, a_{n-1}, \cdots, a_1, a_0]$$

这里，分子、分母多项式系数向量中的系数均按 s 的降幂排列。在 MATLAB 中，可以用函数 tf(Transfer Function)来建立系统的传递函数模型，其基本调用格式为：

```
sys= tf(num,den)
```

其中，sys=tf(num,den)生成连续时间系统的传递函数模型。num 和 den 分别为传递函数

分子和分母的系数向量,且系数必须按照 s 的降幂排列。返回 sys 为函数生成的 tf 对象。

【例 6.5】 已知连续系统传递函数为:

$$G(s)=\frac{4s+2}{2s^4+3s^3+4s^2+5s+6}$$

用 MATLAB 表示出该系统的传递函数模型。

利用 MATLAB 将上述系统模型表示出来的程序和运行结果如下:

```
num = [4 2];
den = [2 3 4 5 6];
G = tf(num,den)
```

执行上述语句后得到:

```
Transfer function:
          4 s + 2
-----------------------------------------
2 s^4 + 3 s^3 + 4 s^2 + 5 s + 6
```

2. 离散系统脉冲传递函数

对于单输入单输出(简称 SISO)的离散时间系统,对公式(6.2)等号两边取 z 变换,就可得到该离散系统的脉冲传递函数为:

$$G(z)=\frac{Y(z)}{R(z)}=\frac{b_m z^m+b_{m-1}z^{m-1}+\cdots+b_1 z+b_0}{a_n z^n+a_{n-1}z^{n-1}+\cdots+a_1 z+a_0} \tag{6.5}$$

在 MATLAB 语言中,输入离散系统的脉冲传递函数模型和连续系统的传递函数模型一样,只需分别定义输入系统的分子、分母多项式系数向量就可以方便地对其加以描述。例如对于公式(6.5),可以先分别定义传递函数的分子、分母多项式系数向量为:

$$\text{num}=[b_m,b_{m-1},\cdots,b_1,b_0]$$

$$\text{den}=[a_n,a_{n-1},\cdots,a_1,a_0]$$

这里,分子、分母多项式系数向量中的系数均按 z 的降幂排列。在 MATLAB 中,同样可以用函数 tf 来建立离散系统的脉冲传递函数模型,其基本调用格式为:

sys= tf(num,den,'Ts',T)

其中,num 和 den 分别为 z 传递函数分子和分母的系数向量,T 为采样周期。返回 sys 为离散系统脉冲传递函数模型。

【例 6.6】 已知离散系统脉冲传递函数模型为:

$$G(z)=\frac{6z^2+2z+1}{3z^4+4z^2-5z+6}$$

且已知系统的采样周期为 $T=0.1$s,试用 MATLAB 表示出该系统的传递函数模型。

利用 MATLAB 将上述系统模型表示出来的程序和运行结果如下:

```
num = [6 2 1];
den = [3 0 4 -5 6];
G = tf(num,den,'Ts',0.1)
```

执行上述语句后得到:

```
Transfer function:
    6 z^2 + 2 z + 1
------------------------------
3 z^4 + 4 z^2 - 5 z + 6
Sampling time: 0.1
```

6.2.4 系统状态方程模型

系统状态方程(SS)是现代控制理论中描述系统模型的一种形式,是研究系统的最为有效的系统数学描述。由于这种描述方式是基于系统内部的状态变量,所以又称为系统的内部描述法,而微分方程和传递函数只能反映出系统输入和输出之间的对应关系,通常称为系统外部模型。

不论是单输入单输出(简称 SISO)系统还是多输入多输出(简称 MIMO)系统,若可用一组一阶微分方程表示,在引进相应的状态变量后,可将这一组一阶微分方程写成状态空间表达式。通常一个线性定常系统(LTI)的状态方程可以表示为:

$$\begin{aligned}\dot{\boldsymbol{X}} &= \boldsymbol{AX} + \boldsymbol{BU} \\ \boldsymbol{Y} &= \boldsymbol{CX} + \boldsymbol{DU}\end{aligned} \tag{6.6}$$

其中,$\boldsymbol{X}$ 为 n 维状态向量;$\boldsymbol{U}$ 为 r 维输入向量;$\boldsymbol{Y}$ 为 m 维输出向量;$\boldsymbol{A}$ 为 $n\times n$ 维的系统状态阵,由系统的参数所决定;$\boldsymbol{B}$ 为 $n\times r$ 维的系统输入阵;$\boldsymbol{C}$ 为 $m\times n$ 维输出变量系数矩阵;$\boldsymbol{D}$ 为 $m\times r$ 维直接传递矩阵。

对于离散时间系统,其状态空间表达式可以写成:

$$\begin{aligned}\boldsymbol{X}(k+1) &= \boldsymbol{FX}(k) + \boldsymbol{GU}(k) \\ \boldsymbol{Y}(k+1) &= \boldsymbol{CX}(k) + \boldsymbol{DU}(k)\end{aligned} \tag{6.7}$$

在 MATLAB 中,用函数 ss 建立相应的系统状态空间模型,其调用格式为:

$$\mathrm{sys}=\mathrm{ss}(A,B,C,D)\text{或者 }\mathrm{sys}=\mathrm{ss}(F,G,C,D,T)$$

其中,A、B、C、D 为系统状态方程系数矩阵;F、G、C、D 为离散系统状态方程系数矩阵;T 为离散系统采样周期。

【例 6.7】 用状态空间模型描述下面的两输入两输出系统:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 2 & 7 & 8 & 9 \\ 4 & 9 & 6 & 8 \\ 2 & 0 & 4 & 7 \\ 7 & 8 & 9 & 7 \end{bmatrix} x + \begin{bmatrix} 7 & 4 \\ 5 & 6 \\ 9 & 8 \\ 1 & 7 \end{bmatrix} u$$

$$\boldsymbol{y} = \begin{bmatrix} 1 & 1 & 3 & 2 \\ 9 & 1 & 6 & 6 \end{bmatrix} x$$

利用 MATLAB 将上述系统模型表示出来的程序和运行结果如下:

```
A = [2 7 8 9;4 9 6 8;2 0 4 7;7 8 9 7];
B = [7 4;5 6;9 8;1 7];
C = [1 1 3 2;9 1 6 6];
D = zeros(2,2);
```

利用上面建立的系数矩阵就可以在 MATLAB 中通过 $G=\mathrm{ss}(A,B,C,D)$来表示该系统

的状态空间模型了。

6.2.5 系统的零极点模型

系统的传递函数还可表示成另一种形式，即零极点形式，其原理是对原系统传递函数的分子和分母多项式进行分解因式处理，以获得系统的零极点表达形式。系统的零极点模型一般可表示为：

$$G(s)=K\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)} \tag{6.8}$$

其中，$z_i(i=1,2,\cdots,m)$和$p_j(j=1,2,\cdots,n)$分别为系统的零点和极点，z_i、p_j既可以是实数也可以是复数；K为系统增益。

在 MATLAB 中，可以使用函数 zpk 来直接建立系统模型的零极点形式，其基本调用格式为：

$$\text{sys}=\text{zpk}(z,p,k) \text{ 或者 } \text{sys}=\text{zpk}(z,p,k,T)$$

其中，z、p、k分别为系统的零极点向量和增益；T为离散系统的采样周期。

另外，MATLAB 还提供了函数 roots 求系统的零极点，调用格式为：

z＝roots(num)　或者　p＝roots(den)

其中，num 和 den 分别为传递函数模型的分子和分母多项式系数向量。

【例 6.8】 已知连续系统传递函数为：

$$G(s)=\frac{4(s+2)(s-1)}{(s+6)(s+5)(s-3)}$$

试用 MATLAB 表示出该系统零极点增益传递函数模型。

利用 MATLAB 将上述系统用零极点模型表示出来的程序和运行结果如下：

```
z = [ - 2 1];
p = [ - 6  - 5 3];
k = 4;
G = zpk(z,p,k)
```

执行上述语句后得到：

```
Zero/pole/gain:
  4 (s + 2) (s - 1)
------------------------
(s + 6) (s + 5) (s - 3)
```

6.2.6 系统模型的转换

在实际的工程应用中，由于要解决的控制系统问题所需要的数学模型和该问题所给定的已知数学模型往往不一致，这时就要将已知的系统数学模型转换成仿真程序能够处理的模型形式。

对系统的数学模型描述主要有传递函数模型、零极点模型和状态空间模型等形式，而这些模型之间又存在着内在的等效关系。在一些场合下用其中的一种模型来处理问题比较方便，而在别的场合下可能又需要另外形式的模型，因此进行模型之间的相互转换是十分必

要的。

1. 系统模型向传递函数模型转换

MATLAB 中传递函数模型的转换主要由下面函数实现,其基本格式为:

```
[num,den] = ss2tf(A,B,C,D,iu)
   [num,den] = zp2tf(z,p,k)
       tfsys = tf(sys)
```

A、B、C、D 矩阵代表系统的状态方程模型,其中 iu 用于指定变换所使用的输入量,对于单输入系统来说,iu=1。对于多变量系统,不能用此函数一次地求出对所有输入信号的整个传递函数矩阵,必须对各个输入信号逐次地求取传递函数子矩阵,最后获得整个传递函数矩阵。第 2 个函数中 z、p、k 为系统的零点、极点和增益矩阵。第 3 个函数可以将任意的对象模型转换成传递函数模型,其中 sys 可以是 ss 模型或 zpk 模型,tfsys 是转换后对应的 tf 模型。

2. 系统模型向状态方程模型转换

MATLAB 系统状态方程模型的转换主要由下面两个函数实现,其基本格式为:

```
[A,B,C,D] = tf2ss(num,den)
  [A,B,C,D] = zp2ss(z,p,k)
       sssys = ss(sys)
```

系统的传递函数模型由 num 和 den 所给出的分子和分母多项式系数来表示,调用上面函数后会自动返回系统的状态方程模型。z、p、k 同样为系统模型的零点、极点和增益矩阵。第 3 个函数可以将系统模型的非状态方程形式转化为状态方程模型。

由传递函数模型求状态空间模型时,所转换得到的模型不是唯一的。因为传递函数只是描述系统输入和输出的关系,这种关系从整体上来看是唯一的,而状态空间是描述系统输入、输出和状态之间的关系,如果状态变量选择不同,状态空间的形式也不同,也即转化后的 ss 模型也就不同。所以第 3 个函数中由传递函数来求系统状态空间的模型,这个实现不是唯一的。

3. 系统模型向零极点模型转换

系统的零极点模型实际上是系统传递函数模型的另一种变形,也是对系统进行分析的一类常用模型。在 MATLAB 中,也提供了将各类系统模型转换为零极点形式模型的函数。

其基本格式为:

```
[z,p,k] = ss2zp(A,B,C,D, iu)
  [z,p,k] = tf2zp(num,den)
       zpsys = zpk(sys)
```

上述第 1 个函数是将以状态方程形式给出的模型根据指定的输入转换为零极点模型形式;第 2 个函数是将传递函数形式给出的模型转换为零极点形式;第 3 个函数可将系统模型的非零极点形式转换为零极点系统模型。

【例 6.9】 已知连续系统传递函数为：

$$G(s) = \frac{2s^2 + 4s + 1}{5s^4 + 4s^2 - 3s + 2}$$

试用 MATLAB 将该系统的传递函数模型转换为状态方程模型和零极点增益模型。

利用 MATLAB 将上述系统模型进行转换的程序和运行结果如下：

```
num = [2 4 1];
den = [5 0 4 - 3 2];
G = tf(num,den);
[A,B,C,D] = tf2ss(num,den)
[z,p,k] = tf2zp(num,den)
zpsys = zpk(G)
```

执行上述语句后得到：

```
A =       0    - 0.8000     0.6000    - 0.4000
     1.0000          0          0           0
          0     1.0000          0           0
          0          0     1.0000           0
B =   1
      0
      0
      0
C =   0     0.4000     0.8000     0.2000
D =   0
z =  - 1.7071
     - 0.2929
p =  - 0.4073  +  0.9665i
     - 0.4073  -  0.9665i
     0.4073  +  0.4446i
     0.4073  -  0.4446i
k =  0.4000
Zero/pole/gain:

            0.4 (s + 1.707) (s + 0.2929)
 ------------------------------------------------------------
(s^2  -  0.8147s  +  0.3636) (s^2  +  0.8147s  +  1.1)
```

4. 连续系统和离散系统模型转换

MATLAB 的控制系统工具箱中提供了连续系统离散化函数 c2d 和离散系统连续化函数 d2c。c2d 的主要功能是将连续系统模型转换为离散时间系统模型，其调用格式为：

sysd=c2d(sysc,Ts,method)

其中，sysc 为连续系统数学模型；sysd 为离散系统数学模型；Ts 为离散系统的采样时间；method 为系统离散模型采用的转换方法，可以使用以下几种方法。

'zoh'为零阶保持器。

'foh'为一阶保持器。

'tustin'为双线性变换方法(Tustin 算法)。

'prewarp'为改进的 Tustin 变换方法。

'matched'为零极点匹配法。

当缺省时默认采用零阶保持器法进行转换。

将离散化系统连续化可以使用 d2c 函数,其调用格式为:

sysc=d2c(sysd,method)

其用法与 c2d 相似,但在调用语句中无需再声明离散系统的采样周期,因为该信息已经包含在离散模型 sysd 中。

【例 6.10】 假设连续系统的传递函数为:

$G(s)=\frac{s+1}{s^2-4s+2}e^{-2s}$,选择采样周期为 $T=0.1$s,若采用 Tustin 算法,求其离散化系统模型。

利用 MATLAB 将上述系统模型进行离散化的程序和运行结果如下:

```
csys = tf([1 1],[1 -4 2])
Transfer function:
    s + 1
-------------------
s^2 - 4 s + 2
csys.ioDelay = 2
Transfer function:
                s + 1
exp(-2*s) * ------------------------
            s^2 - 4 s + 2
dsys = c2d(csys,0.1,'tustin')
Transfer function:
          0.06522 z^2 + 0.006211 z - 0.05901
z^(-20) * --------------------------------------------------
                z^2 - 2.472 z + 1.497
Sampling time: 0.1
```

6.2.7 系统模型的连接

在一般情况下,控制系统的结构往往是由两个或更多个环节或子系统按一定方式连接组合而成的,它们之间的连接方式有并联、串联、反馈等。要对连接而成的系统进行分析,就需要对系统的模型进行适当的处理。MATLAB 控制系统工具箱中提供了大量的函数来对控制系统的简单模型进行连接。

1. 模型串联

函数 series 用于两个线性模型串联,其调用格式为:

sys = series (sys1, sys2)

图 6.11 是一个单输入单输出(SISO)串联模型,sys1、sys2 和 sys 的关系如图 6.11 所示。

u(t) → sys1 → sys2 → y(t)

图 6.11 串联结构图

该函数的执行结果等价于 sys = sys1 * sys2。

对于多输入多输出(MIMO)系统,函数 series 的调用格式为:

sys = series (sys1, sys2, outputs1, inputs2)

函数执行系统 sys1 和系统 sys2 串联时,将系统 sys1 的输出端 1 和 sys2 的输入端 2 连接。

2. 模型并联

函数 parallel 用于两个模型并联,其调用格式为:

sys = parallel (sys1, sys2)

其中,sys1、sys2 和 sys 关系如图 6.12 所示。

该函数执行结果等价于 sys = sys1 + sys2。

对于 MIMO 系统,函数 parallel 的调用格式为:

sys = parallel (sys1, sys2, IN1, IN2, OUT1, OUT2)

函数执行系统 sys1 和 sys2 并联时,将 sys1 的输入端 IN1 和 sys2 的输入端 IN2 连接,sys1 的输出端 OUT1 和 sys2 的输出端 OUT2 连接起来。

3. 反馈连接

函数 feedback 用于模型的反馈连接,其调用格式为:

sys = feedback (sys1, sys2, sign)

其中,sys1、sys2 和 sys 如图 6.13 所示。sign 用于定义反馈形式,当采用负反馈时,sign 可忽略,也可以为−1;当采用正反馈时,sign = +1。

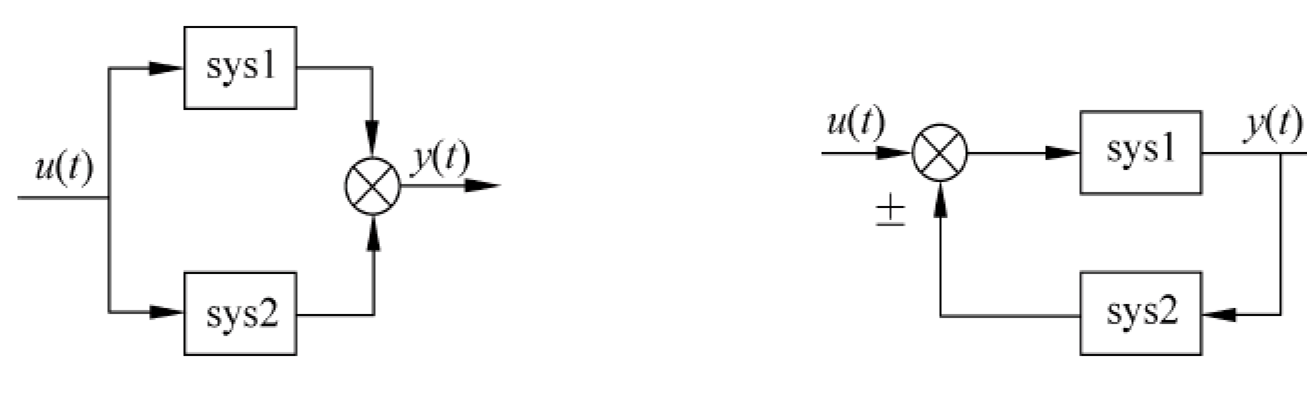


图 6.12 并联结构图　　图 6.13 反馈结构图

【例 6.11】 假设有两个模型 $G_1(s)=\frac{2s-1}{s^2+3s+1}$ 和 $G_2(s)=\frac{s+2}{(s+3)(s+4)}$,如果采用负反馈结构可以用下面的 MATLAB 语句得到整个系统的传递函数模型。

```
G1 = tf([2 -1],[1 3 1]);
G2 = tf([1 2],[1 7 12]);
G = feedback(G1,G2)
Transfer function:
      2 s^3 + 13 s^2 + 17 s - 12
------------------------------------------
s^4 + 10 s^3 + 36 s^2 + 46 s + 10
```

若采用正反馈连接结构,则可由下面的语句来实现。

```
G = feedback(G1,G2, +1)
结果为:
Transfer function:
    2 s^3 + 13 s^2 + 17 s - 12
--------------------------------------------
s^4 + 10 s^3 + 32 s^2 + 40 s + 14
```

6.3 控制系统的MATLAB仿真

控制系统的计算机仿真就是以前面建立的控制系统数学模型为基础，采用数学模型代替实际的系统，以计算机作为主要工具，对控制系统进行实验和分析研究的一种方法。

6.3.1 控制系统仿真的过程

1. 建立控制系统的数学模型

系统的数学模型是描述系统的输入和输出变量以及内部各变量之间关系的数学表达式。对于一些系统可以根据其所遵循的物理、化学基本定律(例如牛顿定律、基尔霍夫定律、动力学定律等)来建立数学模型，这也就是所谓的解析法建立数学模型。而对于相对比较复杂的系统，难以写出用数学表达式表示的数学模型，则必须通过实验的方法，通过系统辨识技术来建立数学模型，这就是所谓的实验法建立数学模型。

数学模型是系统仿真研究的主要依据，所以数学模型的准确性是十分重要的。

2. 建立控制系统的仿真模型

一般的数学模型不能直接编制程序对系统进行计算机求解仿真，通常将其转换成能在计算机上对系统进行仿真的模型，也就是要通过一定的算法对系统的数学模型进行处理。对于连续系统，就是将微分方程在零初始条件下进行拉普拉斯变换，求得控制系统的传递函数模型。对于离散系统，就是将差分方程经Z变换转换为计算机可以编程仿真处理的数字控制器模型。

3. 编制仿真程序

对于非实时系统仿真，可以用高级语言(例如Basic、Fortran、C等)依据相应的算法编制系统的仿真程序。而对于实时快速系统的仿真，则主要采用汇编语言或者与高级语言共用的方式进行编程。当然也可以采用专门的仿真语言和仿真软件包。

目前，美国MathWorks公司的动态仿真集成软件Simulink与MATLAB软件集成在一起，用来研究和分析控制系统是非常方便的，成为当今最具影响力的控制系统仿真语言环境。

4. 程序调试、模型验证以及实验结果分析处理

程序调试主要是检查并纠正程序编写的错误，使其能在计算机上正确的运行。模型验证是通过程序的正确运行，用仿真的实验结果与实际系统运行观测的结果相比较，来验证模型是否能代表实际系统，以反映实际系统运行过程的特性。

6.3.2 控制系统的计算机仿真

基于MATLAB的控制系统计算机仿真方法有采用M文件编写程序的形式，或者使用

Simulink 搭建模型的形式。

1. 线性系统时域分析

用 MATLAB 控制系统工具箱中线性系统时域响应分析功能可以方便地进行线性系统时域分析。

【例 6.12】 已知某系统的闭环传递函数为 $G(s)=\dfrac{10}{Ts+1}$，试求 T 分别为 $T=0.1\text{s}$、0.3s 和 0.5s 时的阶跃响应曲线(在同一坐标系中绘出)。

可以用下面的 MATLAB 语句来实现：

```
syms  T
for T = [0.1 0.3 0.5]
    num = [10];
    den = [T 1];
    G = tf(num,den);
    step(G);
    hold on
end
grid on
legend('T = 0.1','T = 0.3','T = 0.5')
```

执行上面的语句后，会在同一坐标系中显示 T 为不同数值时的阶跃响应曲线，其结果如图 6.14 所示。

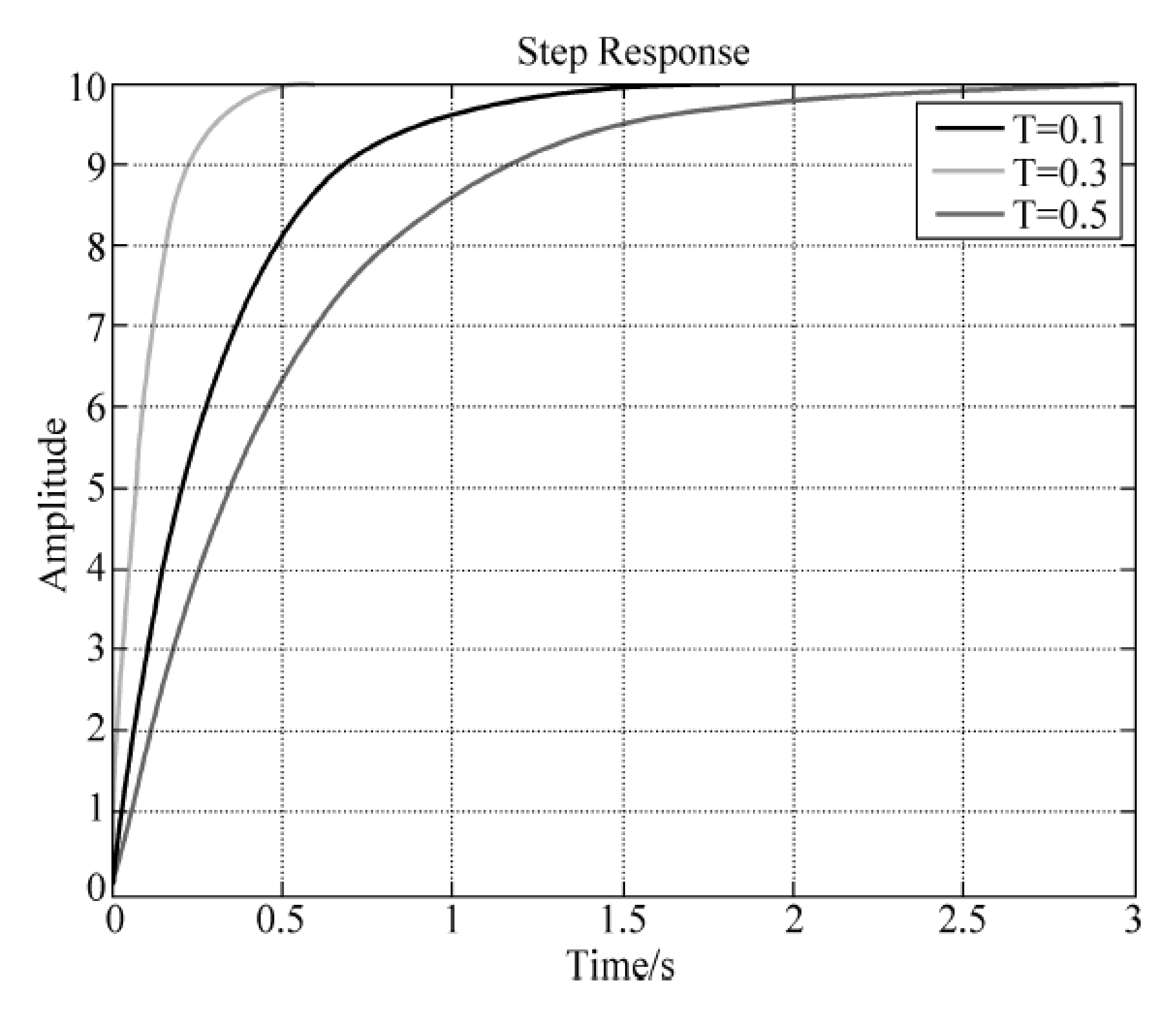


图 6.14 T 为不同数值时的单位阶跃响应曲线

2. MATLAB 在离散系统的分析应用

【例 6.13】 对于如图 6.15 所示的计算机控制系统结构图，已知系统采样周期为 $T=0.1\text{s}$，被控对象的传递函数为 $G(s)=\dfrac{2}{s(0.1s+1)(0.05s+1)}$，数字控制器 $D(z)=\dfrac{z-0.36}{z+0.98}$，试求该系统的闭环脉冲传递函数和单位阶跃响应。

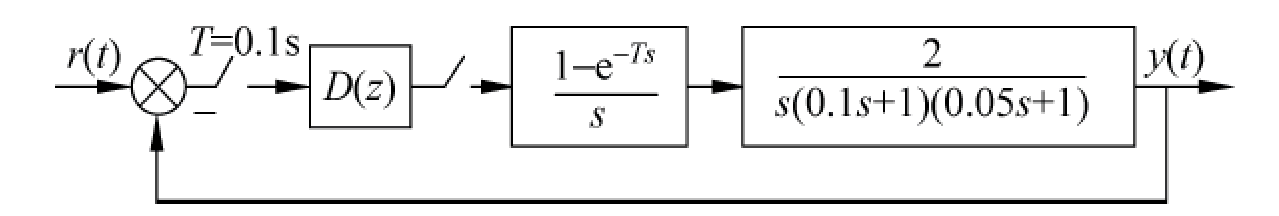


图 6.15　计算机控制系统结构图

首先求解开环脉冲传递函数，可以用下面的 MATLAB 语句来实现。

```
T = 0.1;
sys = tf([2],[0.005 0.15 1 0]);
sys1 = c2d(sys,T,'zoh');
sys2 = tf([1 -0.36],[1 0.98],0.1);
sys3 = series(sys2,sys1)
```

执行后在屏幕上显示系统的开环脉冲传递函数为

```
Transfer function:
0.03362 z^3 + 0.05605 z^2 - 0.01699 z - 0.002717
-----------------------------------------------------------------
z^4 - 0.5232 z^3 - 0.9201 z^2 + 0.4922 z - 0.04879
Sampling time: 0.1
```

然后求其闭环脉冲传递函数，可以输入下列命令来实现。

```
sys4 = tf([1]);
sys5 = feedback(sys3,sys4,-1)
```

执行后在屏幕上显示系统的闭环脉冲传递函数为

```
Transfer function:
0.03362 z^3 + 0.05605 z^2 - 0.01699 z - 0.002717
-----------------------------------------------------------------
z^4 - 0.4896 z^3 - 0.8641 z^2 + 0.4752 z - 0.05151
Sampling time: 0.1
```

最后用下列命令即可绘制该离散系统的单位阶跃响应，其结果如图 6.16 所示。

```
step(sys5)
```

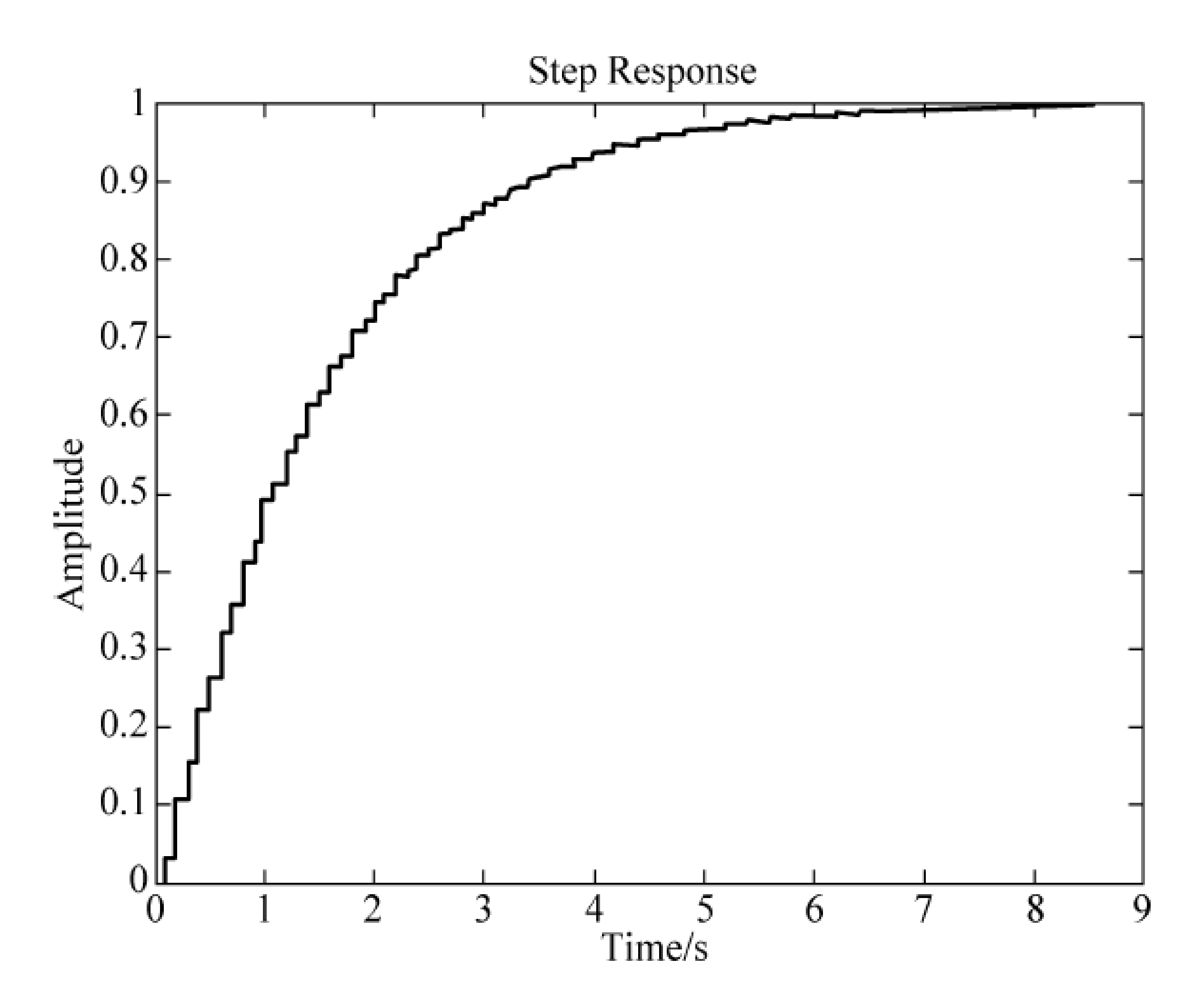


图 6.16　离散系统的单位阶跃响应曲线

3. Simulink 在离散系统的分析应用

例 6.13 所给的离散系统的 Simulink 的仿真模型如图 6.17 所示。在建立的仿真模型中，设置数字控制器和零阶保持器的采样时间为 0.1s，运行仿真模型就可以获得系统的单位阶跃响应。双击打开 Scope 模块，就可以观察到运行结果。同时可以把运行的结果送到 MATLAB 的工作空间，在 MATLAB 的命令窗口执行 plot(tout,y(:,2))命令即可绘制出该离散系统的单位阶跃响应，如图 6.18 所示。

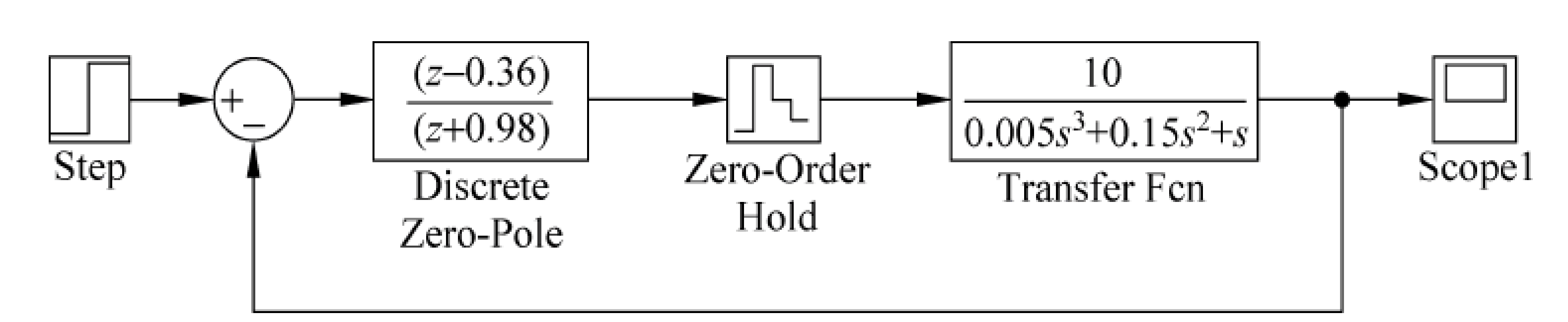


图 6.17　系统的 Simulink 的仿真模型图

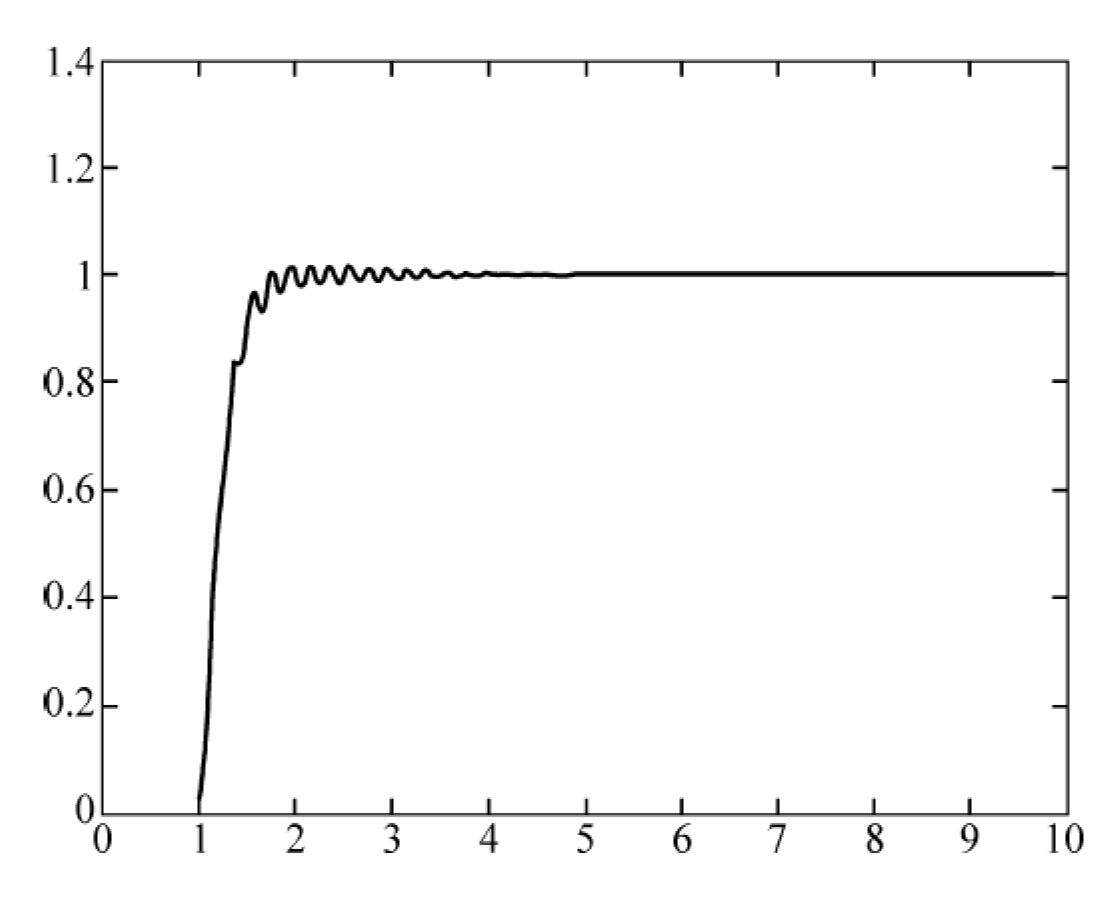


图 6.18　离散系统的单位阶跃响应曲线

【例 6.14】 对于如图 6.19 所示的计算机控制系统结构图，已知系统采样周期为 $T=1\text{s}$，被控对象和零阶保持器的传递函数分别为 $G(s)=\dfrac{10}{s(s+1)}$ 和 $H(s)=\dfrac{1-e^{-Ts}}{s}$，针对单位速度函数输入，为了将系统设计成最少拍无纹波控制系统，其设计的数字控制器为 $D(z)=\dfrac{0.272(1-0.3679z^{-1})(1.408-0.825z^{-1})}{(1-z^{-1})(1+0.592z^{-1})}$，试建立该系统的 Simulink 仿真模型并分析。

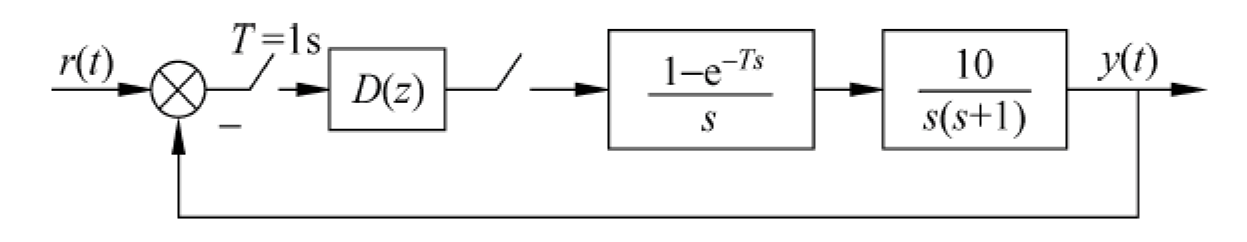


图 6.19　计算机控制系统结构图

在 MATLAB 的 Simulink 窗口，根据系统结构图，到相应的模型库中拖拉对应的模型并修改参数即可完成系统的 Simulink 仿真模型的搭建，其模型如图 6.20 所示。

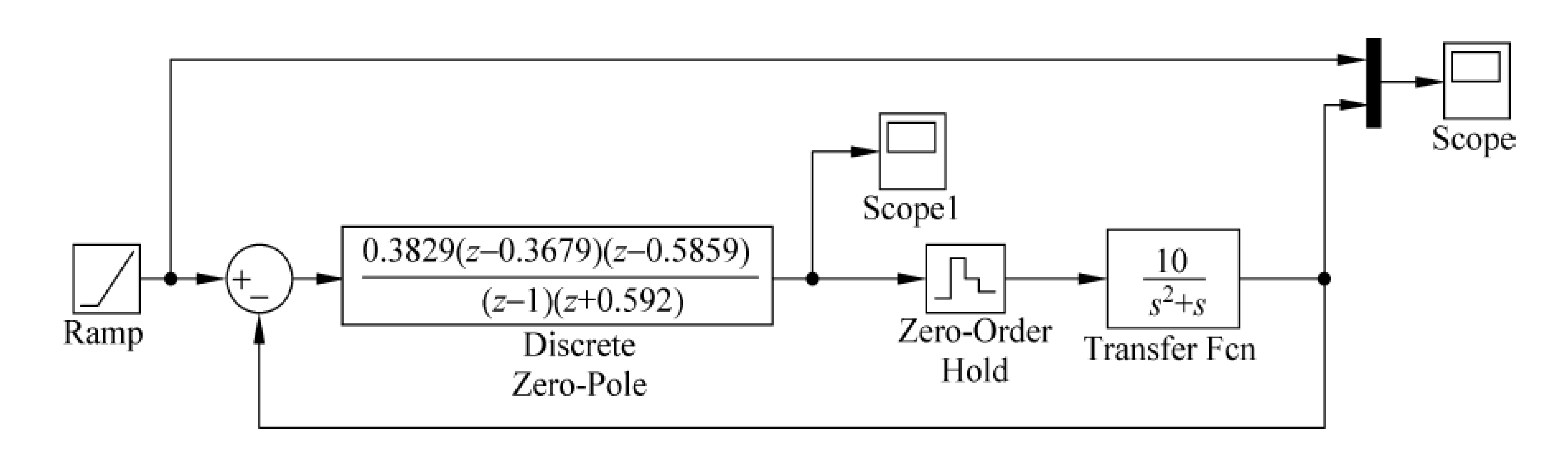


图 6.20　系统的 Simulink 仿真模型图

数字控制器输出如图 6.21 所示。从数字控制器的输出曲线可以看出，数字控制器的输出在第 3 个采样周期之后为常数，所以系统在非采样点之间不再有纹波。

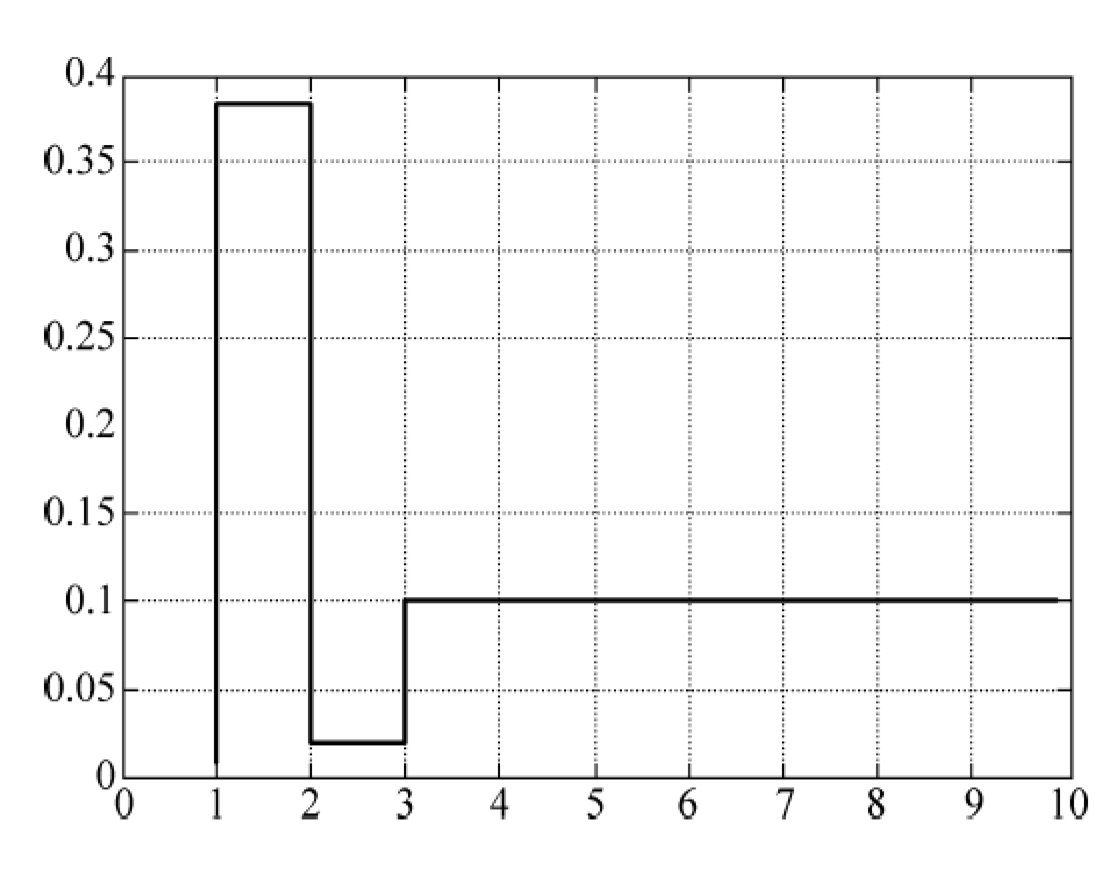


图 6.21　数字控制器输出

设置示波器 Scope 的参数，将其数据以数组形式输入到 MATLAB 的工作空间，并命名为 y，然后在 MATLAB 的命令窗口输入下列命令：

```
plot(y(:,1),y(:,2:3))
grid
legend('输入','输出')
```

系统的输入输出曲线如图 6.22 所示。

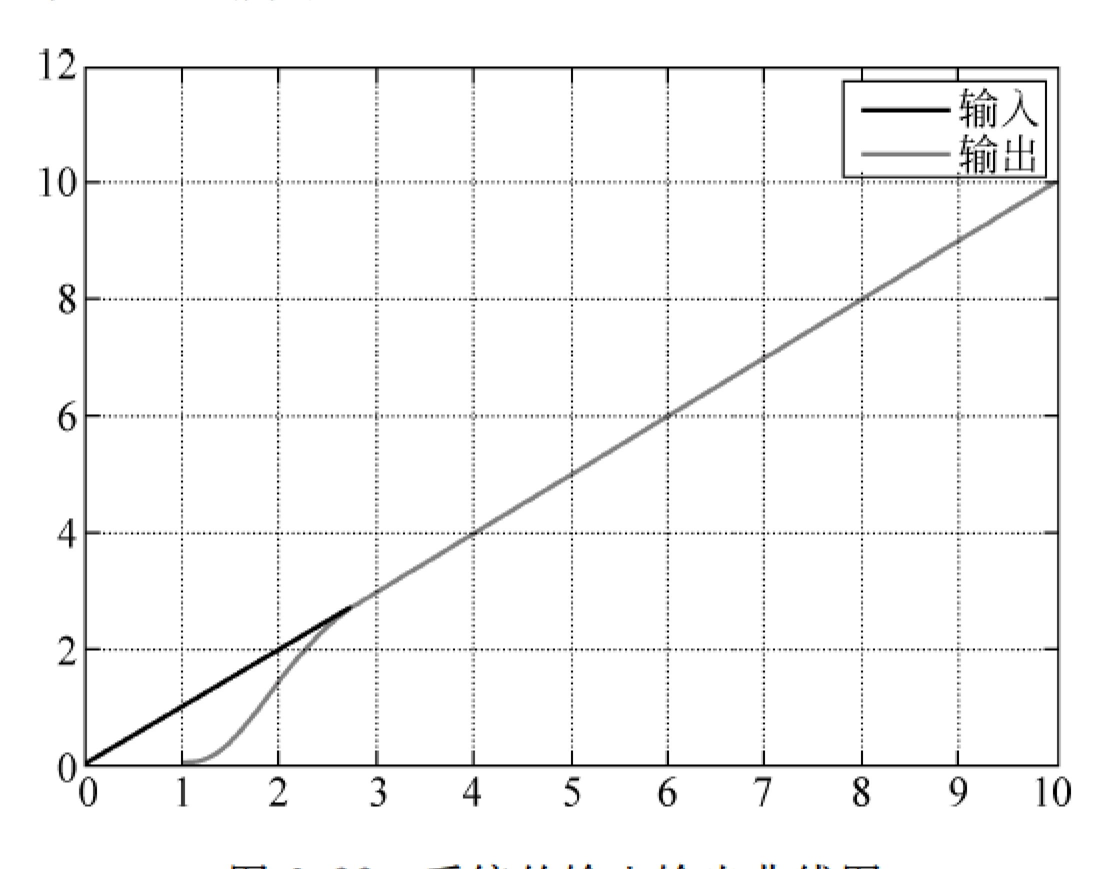


图 6.22　系统的输入输出曲线图

通过分析图 6.22 的仿真结果可知：按最少拍无纹波设计的控制器，其系统的输出值在跟踪输入值之后，在非采样点之间不再有波纹，不仅在采样点跟踪的非常好，在采样点之间也跟踪的非常好。所设计的控制器使系统满足最少拍无纹波控制系统的要求。

习题

1. 简述 MATLAB 语言的主要特点。
2. 简述 MATLAB 桌面平台主要窗口的功能。
3. 比较 MATLAB 函数 M 文件和命令 M 文件的异同。
4. 在同一坐标系中绘制 $y=\sin(x)$ 和 $y=\cos(x)$ 的图形，要求正弦曲线为蓝色、虚线；余弦曲线为红色、点划线。
5. 试简述控制系统建立数学模型的方法，并说明常用的控制系统模型有哪些？各有什么特点？
6. 已知被控对象的传递函数为：$G(s)=\dfrac{6(s+4)}{s^2+2s+9}$，试用 MATLAB 建立该系统的模型。
7. 已知单位负反馈二阶系统的开环传递函数为：$G(s)=\dfrac{5}{s^2+4s}$，建立该系统的 Simulink 仿真模型并绘制单位阶跃响应曲线图。



第7章 数字程序控制技术

数字程序控制系统即计算机数字控制(Computer Numerical Control,CNC)系统,是利用数字电子技术和计算机技术,对生产机械进行自动控制的系统,它能够根据所采集的数据和预先编制好的程序,控制生产机械(如各种加工机械)按相应的工作顺序、运动轨迹、运动距离和运动速度等自动地完成系统的控制。

本章主要介绍:数字程序控制的基本原理和控制方式;逐点比较法直线插补和圆弧插补的原理及程序实现;以步进电机作为驱动设备的数字程序控制技术。

7.1 数字程序控制基础

数字程序控制简称为数控(CNC),它用代表加工顺序、加工方式和加工参数的数字码作为控制指令,主要用于各种机床的自动控制,如铣床、车床、加工中心、线切割机,以及焊接机、气割机的控制等。这种采用数字程序控制的机床叫做数控机床,数控机床正是由于采用了数字程序控制的方式,不仅能够加工形状复杂的零件,而且加工精度高、生产效率高,是实现工业生产自动化的重要保障。同时数控技术和数控机床是实现柔性制造(Flexible Manufacturing,FM)和计算机集成制造(Computer Integrated Manufacturing,CIM)的最重要的基础技术之一。

7.1.1 数控技术的发展

近年来,随着计算机技术的发展,数字控制技术已经广泛应用于工业控制的各个领域,尤其是机械制造业中,普通机械正逐渐被高效率、高精度、高自动化的数控机械所代替。目前国外机械设备的数控化率已达到85%以上,而我国的机械设备的数控化率不足20%。

世界上第一台3坐标立式数控机床是1952年美国麻省理工学院伺服机构实验室开发出来的,主要目的是为了满足高精度和高效率的加工复杂零件的需要。它标志着数控技术发展的开始。在零件的传统加工过程中,二维和三维轮廓零件的加工是很困难的,而有了数控机床后则显得非常容易。数控技术的发展大致分为以下四个阶段。

第一阶段:1952年至1970年,这一阶段的数控系统是以数字电子技术为基础来实现的,经历了三个时代,即电子管时代、晶体管时代和小规模集成电路时代,各种控制功能,如输入装置、插补运算、控制器等,都是由硬件逻辑电路来实现。控制功能比较简单,使用灵活性较差。

第二阶段:1970年至1974年,由于计算机的迅速发展,其性价比不断提高,小型计算机

代替了数控系统中硬件构成的专用计算机装置,从而诞生了计算机数控,简称 CNC 系统。它通过软件实现控制,使数控系统的通用性和灵活性都得到了很大提高。数控系统中的输入装置,输出装置,控制器和插补器等四大部分的功能都可由计算机系统来完成。

第三阶段:1974 年至 20 世纪 80 年代初,随着采用大规模集成电路的微处理器迅速发展,数控系统开始采用微型计算机。由于微处理器的集成度不断提高,运算速度越来越快,功能越来越丰富。但是这种数控系统一直由一些厂家进行封闭垄断性生产,使得这类数控产品专用性强,与标准计算机不兼容,通用性、软件移植性和组网通信能力较差。

第四阶段:20 世纪 90 年代初开始,数控系统向开放式方向发展,具体表现是基于 PC 的开放式数控系统,这种系统从 20 世纪 80 年代初开始提出研制,直到 20 世纪 90 年代初得以实现。它结合了 PC 的分析运算能力、大容量存储功能、图文显示优势及组网的灵活性,使得数控系统的使用具备了较为开放的模式。同时,随着基于 DSP(Digital Signal Processor,数字信号处理器)的运动控制技术的突破,为开放式数控系统的发展创造了新的条件。基于 DSP 的运动控制器为核心,融合 PLC 功能,与标准 PC 集成的新一代开放式数控系统将成为新一代数控系统的主导产品。

随着生产技术的发展,对产品的性能要求越来越高,产品改型频繁,采用多品种小批量生产方式的企业越来越多,这就要求数控机床向着高速度、高精度、复合化、系统化、智能化、环保化方向发展。

7.1.2 数控系统的分类

数控系统按被控制对象的运动轨迹进行分类,可分为点位控制、直线控制和轮廓控制。

(1) 点位控制数控机床的数控装置只要求能够精确地从一个坐标点到另一个坐标点的精度定位,而不管从一点到另一点是按什么轨迹运动,并且在移动过程中不进行任何加工。为了精度定位和提高生产效率,系统首先需要高速运行,然后按 1~3 级减速,使之慢速趋近于定位点,减小定位误差。

这类数控机床主要有数控钻床、数控坐标镗床、数控冲床、数控点焊机和数控折弯机等。

(2) 直线控制的数控机床一般要在各点间移动的同时进行切削加工,所以不仅要求机床有准确的定位功能,还要求从一点到另一点之间按直线规律运动,而且对运动的速度也要进行控制,即对于不同的刀具和工件,需要选择不同的进给速度。

这一类机床包括简易数控车床、数控铣床、数控镗床等。一般情况下,这些机床可以有两到三个可控轴,但一般同时控制轴数只有两个。

(3) 轮廓控制又称连续控制,大多数数控机床具有轮廓控制功能。其特点是能同时控制两个以上的轴,具有插补功能。它不仅控制起点和终点位置,而且要控制加工过程中每一点的位置和速度,可以加工出任意形状的曲线或曲面组成的复杂零件。

轮廓控制的数控机床包括有两坐标及两坐标以上的数控铣床、可以加工回转曲面的数控机床、加工中心等。

数控系统按控制方式分类,可分为开环控制系统、闭环控制系统和半闭环控制系统。

(1) 开环控制系统没有检测反馈装置,数控装置发出的指令信号流程是单向的,其精度主要决定于驱动元件和伺服电机的性能。开环数控机床系统所用的驱动设备主要是步进电动机,移动部件的速度与位移是由输入脉冲的频率和脉冲数决定,位移精度主要决定于该系

统各有关零部件的制造精度。系统框图如图 7.1 所示。

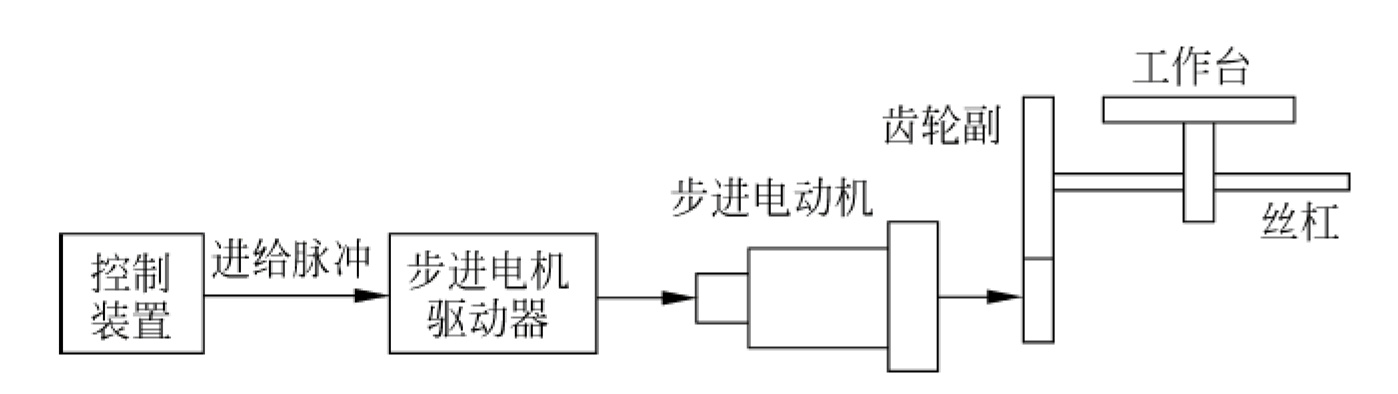


图 7.1　开环数控系统结构框图

开环控制系统具有结构简单、系统稳定、容易调试、成本低等优点。但是系统对移动部件的误差没有补偿和校正，所以精度低，一般位置精度通常为±0.01～±0.02mm。一般适用于经济型数控机床和旧机床数控化改造。

(2) 闭环控制系统是指在机床的运动部件上(如工作台)安装位置测量装置(如光栅、感应同步器等)，系统运行过程中可以及时将运动部件的实际位置反馈到控制装置中，与输入的期望位置相比较，从而实现移动部件的最终精确定位，如图 7.2 所示。从理论上讲，闭环控制系统的控制精度主要取决于检测装置的精度，它完全可以消除由于传动部件制造中存在的误差给工件加工带来的影响，所以这种控制系统可以得到很高的加工精度。闭环控制系统主要用于一些精度要求较高的镗铣床、超精车床和加工中心等。

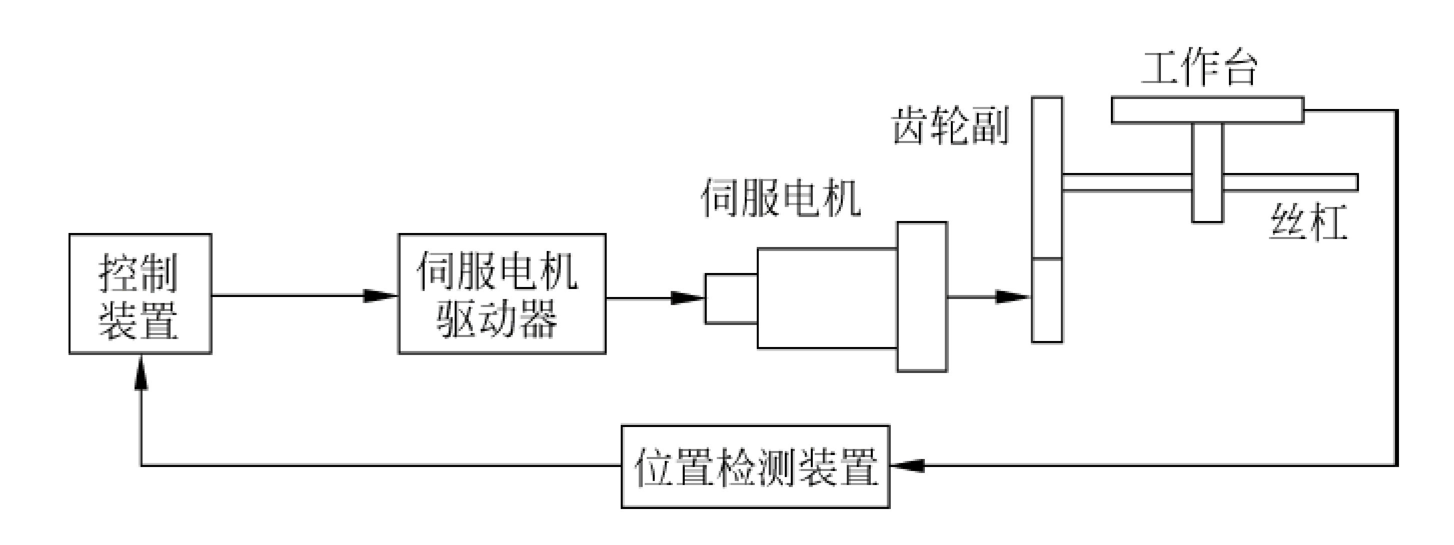


图 7.2　闭环数控系统结构框图

(3) 半闭环控制系统是在开环系统的丝杠上或进给电动机的轴上装有角位移检测装置，如圆光栅、光电编码器或旋转式感应同步器等，通过检测丝杠转角或电机的转角间接地测量工作台位移量，然后反馈给控制装置，不是直接测量工作台的位移量。由于反馈只能对反馈环以内的误差和扰动起到调整作用，对反馈环以外的如丝杠的螺母副的传动误差等无法测量和调整。因而半闭环控制系统的精度低于闭环系统。但与闭环数控系统相比，由于角位移检测装置比直线位移检测装置结构简单且安装调试方便，因此配有精密滚珠丝杠和齿轮的半闭环系统正在被广泛地采用，如图 7.3 所示。

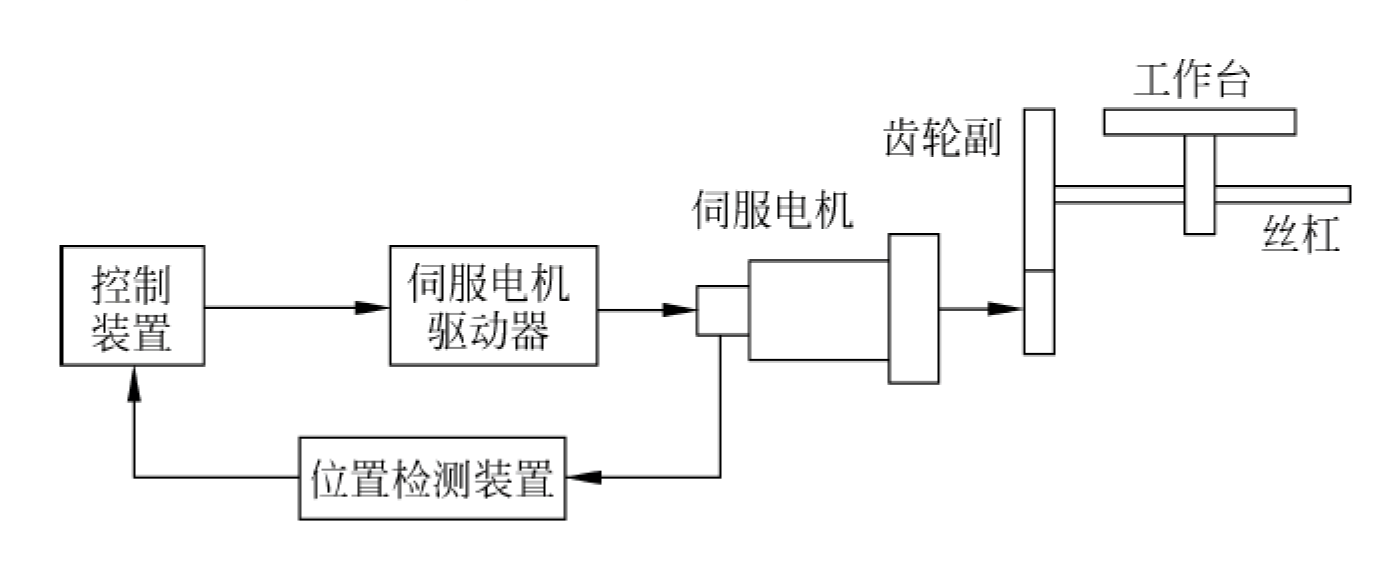


图 7.3　半闭环数控系统结构框图

7.1.3 数字程序控制原理

数字程序控制系统(数控系统)一般由输入装置、输出装置、控制器、插补器、伺服驱动装置等几部分组成。主要作用就是将进给位移量等信息转换成机床的进给运动,使系统正确、快速地跟随控制信息,执行机械运动;同时,位置反馈系统将机械运动的实际位移信息反馈至数控系统,以确保较高的控制精度。

1. 数字程序控制的主要任务

从数控机床的控制任务来看,主要包括主轴驱动、进给运动两个方面。

1) 主轴驱动

主轴驱动主要完成切削任务,其动力约占整台机床动力的70%～80%。基本控制方式是主轴的正转、反转、停止,以及自动换挡和无级调速;对加工中心和有些数控车床还必须具有定向控制和C轴控制。

2) 进给运动

进给运动是数控机床区别于普通车床最根本的地方,它用电气驱动替代了机械驱动。数控机床的进给运动是由进给伺服系统完成的。伺服系统包括伺服驱动装置、伺服电动机、进给传动链及位置检测装置等。伺服控制主要就是机床工作台或刀具的位置控制,伺服系统中所采取的一切措施,都是为了保证进给运动的定位精度。如采用高精度的位置检测装置,采用高性能的伺服驱动装置和伺服电动机,提高数控系统的运算速度等。本章主要针对步进电机作为驱动器的轨迹控制进行介绍。

2. 插补的基本概念

数控系统中的轨迹控制策略是插补和位置控制。

什么是插补呢?在数控机床中,刀具不能严格地按照要求加工的曲线运动,只能用折线轨迹逼近所要加工的曲线,即根据已知曲线的曲线类型(如直线、圆弧或高次曲线)、起点、终点以及速度,按照某种算法计算已知点之间的中间点,从而确定刀具运动轨迹的过程,也称为“数据点的密化”,这就是插补(Interpolation)。

数控装置根据输入的加工零件程序的信息,将程序段所描述的曲线的起点、终点之间的空间进行数据密化,从而形成要求的轮廓轨迹,并且边计算边根据计算结果向各坐标轴的驱动设备发出进给脉冲,对应每个脉冲,机床在响应的坐标方向上移动一个脉冲当量的距离,从而将工件加工出所需要轮廓的形状。当然,单轴运动就不存在插补问题。

数控加工轮廓一般由直线、圆弧组成,有时也有一些非圆的曲线轮廓,因此可以分段进行轮廓线的拟合插补。如果驱动装置为步进电机,驱动每个轴以一定步长运动,实现以折线轮廓拟合光滑曲线轨迹的加工。如图7.4所示即为用插补的方法,用折线拟合从点(0,0)到点(6,4)直线的过程。

3. 数控系统的一般工作过程

数控系统的一般工作过程可分为 4 个基本步骤，如图 7.5 所示。

(1) 首先根据所要加工的工件图纸进行数控加工程序的编写，可直接在数控设备自带的编程设备上编写，也可以在个人计算机上编写后通过相应的通信接口输入给数控设备。同时将相关控制参数和补偿数据(如刀具补偿)等输入给数控系统。

(2) 对程序进行译码和预处理。

(3) 插补运算处理。

(4) 伺服控制。将计算机送出的位置进给脉冲或进给速度指令，经变换和放大后转化为伺服电机(步进电机或交、直流伺服电机)的转动，从而带动工作台移动。

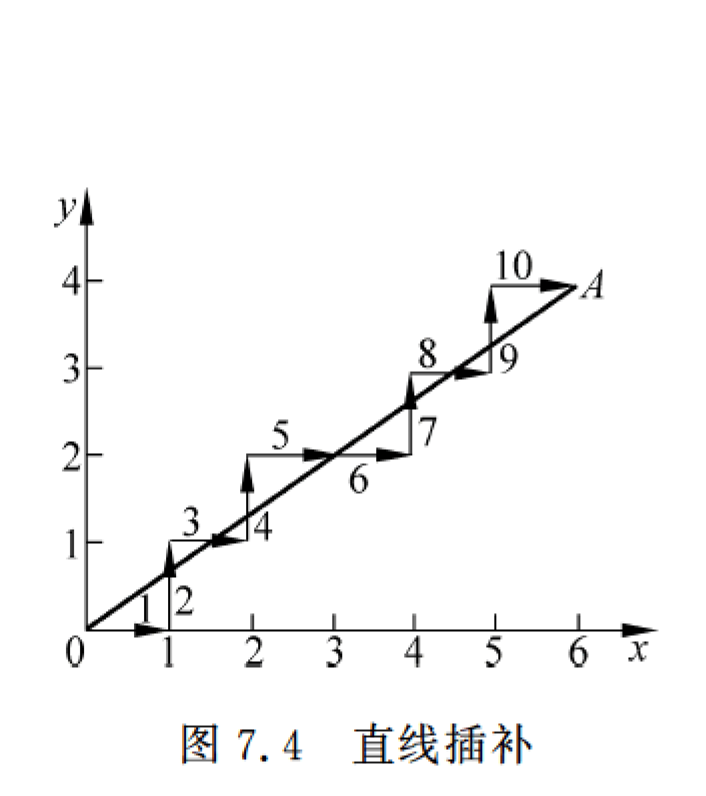


图 7.4　直线插补

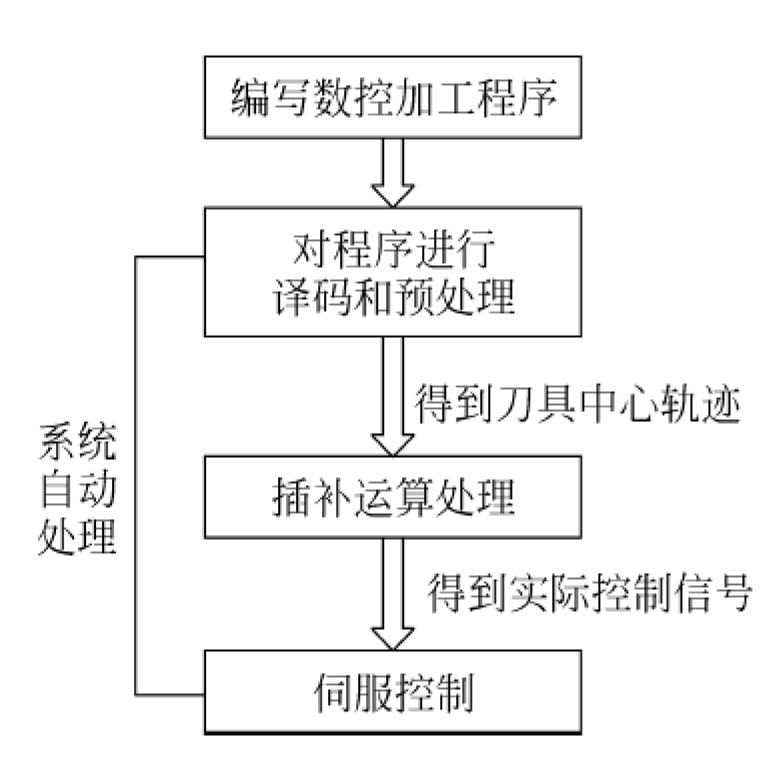


图 7.5　数控系统的一般工作过程

在轮廓加工中，当采用不同尺寸的刀具加工同一轮廓工件，或虽是同一刀具，但因磨损而使其尺寸变化时，为了保证控制精度，都需进行刀具补偿，以及补偿参数的调整。

7.2　逐点比较法插补原理

插补模块是数控系统中一个及其重要的功能模块，它将直接影响到数控系统的运动精度、运动速度和加工能力等。插补运算可以采用数控系统硬件或数控系统软件来完成。即有硬件插补器和软件插补器之分。二者比较如下。

- 硬件插补器：速度快，但缺乏柔性，调整和修改都困难。
- 软件插补器：速度慢，但柔性高，调整和修改都很方便。

早期的数控系统采用由数字逻辑电路组成的硬件插补器，随后出现了很多专门用于插补的集成电路，如 MCX314A 芯片内部就含有高速高精度的直线和圆弧插补功能。目前的 CNC 系统多采用软件插补器，或软件、硬件相结合的插补方式。

7.2.1　插补算法的分类

目前常用的插补方法大致分为两类：脉冲增量插补和数字增量插补。

脉冲增量插补，主要用于采用步进电机驱动的开环系统。每次插补计算结束，CNC 装置向各坐标轴驱动装置发出一个脉冲，每个脉冲代表了一个单位的行程增量，这个最小位移称为脉冲当量，驱动步进电机带动机床移动部件运动。脉冲序列的频率代表了坐标运动速度，而脉冲的数量表示移动量。其基本思想是用折线来逼近直线和曲线。这类插补算法有逐点比较法、最小偏差法、数字积分法等。适用于一些中等精度或中等速度要求的计算机数控系统。

数字增量插补，又称为时间标量插补或数据采样插补。主要用于采用交、直流伺服电机为伺服驱动系统的闭环、半闭环数控系统，也可以用于以步进电机为伺服驱动系统的开环数控系统。其特点是 CNC 装置产生的不是单个脉冲，而是标准的二进制数。其基本思想是用直线段来逼近曲线(包括直线)。

采用数字增量插补时，插补程序以一定的时间间隔定时进行。根据编程的速度将轮廓曲线分割为插补采样周期的进给段即轮廓步长，用弦线和割线逼近轮廓轨迹。在每一插补周期内，插补程序被调用一次，计算出各坐标轴在下一插补周期内的位移增量(数字量而不是单个脉冲)ΔX、ΔY 等，然后再计算出相应插补点位置的坐标值。插补运算速度与进给速度无严格的关系，因此可以达到较高的进给速度。数字增量插补的实现算法较脉冲增量插补复杂，对计算机的运算速度有一定要求。这类插补算法有数字积分法、二阶近似插补法、时间分割法等。

7.2.2 逐点比较法插补算法

脉冲增量插补算法包括有逐点比较法、最小偏差法、数字积分法等。比较常用的是逐点比较法。逐点比较法的基本原理是：数控系统在控制加工过程中，逐点计算和判别加工误差，并与规定的运动轨迹进行比较，由比较结果决定下一步的移动方向。这种算法的特点如下。

(1) 运算简单、直观，可以实现直线插补和圆弧插补。

(2) 每次插补运算后，只有一个坐标轴方向有进给。

(3) 插补误差小于一个脉冲当量，因此只要把脉冲当量(每走一步的距离即步长)取得足够小，就可达到加工精度的要求。

(4) 输出脉冲均匀，输出脉冲的速度变化小，调节方便。

缺点是不容易实现两坐标以上的联动插补。因此在两坐标联动的数控机床中应用比较普遍。

直线和圆弧是构成零件轮廓的基本线型，所以绝大多数数控系统都具有直线插补和圆弧插补功能。本章重点介绍直线插补和圆弧插补的计算方法。

1. 逐点比较法直线插补

逐点比较法直线插补就是数控系统的刀具或绘图笔每走一步都要和给定轨迹上的坐标值进行比较，看这点在给定轨迹的上方或下方，从而决定下一步的进给方向，如图 7.4 所示。如果当前点在给定轨迹的下方，下一步就向给定轨迹的上方走；如果当前点在给定轨迹的

上方，下一步就向给定轨迹的终点方向走。如此走一步，比较一次，决定下一步走向，用折线来逼近给定轨迹，即形成逐点比较插补。

逐点比较法直线插补的工作过程的每一步都要经过以下4个工作节拍。

(1) 偏差判别：根据偏差值的符号，判别刀具当前位置相对于零件轮廓的位置偏差。

(2) 坐标进给：根据偏差判别的结果，控制相应的坐标轴进给一步，使刀具向零件轮廓靠拢。

(3) 偏差计算：刀具进给一步后，针对新的刀具位置，计算新的偏差值。

(4) 终点判断：刀具进给一步后，需要判别刀具是否已经到达零件轮廓的终点。如果已经到达终点，则停止插补过程；如果未到达终点，则返回到第(1)步，重复上述4个节拍。下面以第一象限直线为例介绍插补过程。

1) 位置偏差的计算

设有第一象限直线OA，如图7.6所示，起点O为坐标原点，终点为$A(x_e, y_e)$，坐标系中各点坐标的单位为脉冲当量数。

假设点$m(x_m, y_m)$为加工点（动点），若点m在直线段OA上，则：

$$\frac{y_m}{x_m} = \frac{y_e}{x_e}$$

即：$y_m \cdot x_e - x_m \cdot y_e = 0$，则定义直线插补的偏差判别式为$F_m = y_m \cdot x_e - x_m \cdot y_e$。

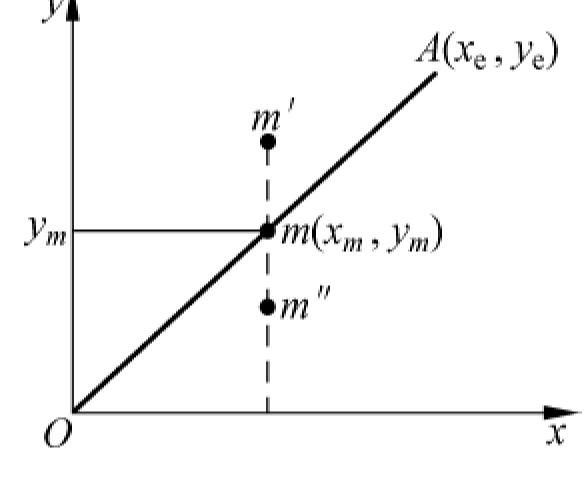


图7.6　第1象限直线OA

若$F_m = 0$，表明加工点m在OA直线段上。

若$F_m > 0$，表明加工点到原点斜率偏大，点m在OA直线段的上方，即点m'处。

若$F_m < 0$，表明加工点m在OA直线段的下方，即点m''处。

由此可得第一象限直线逐点比较法插补的原理是：从直线的起点O（即坐标原点）出发，当$F_m > 0$时，动点在直线上方，沿$+x$轴方向进给一步；当$F_m < 0$时，动点在直线下方，沿$+y$轴方向进给一步；当$F_m = 0$时，动点在直线上，既可以向$+x$方向也可以向$+y$方向进给一步，在此约定取$+x$方向。当两方向所走的步数与终点坐标(x_e, y_e)相等时，发出到达终点信号，停止插补。

但是，上述位置误差F_m的计算公式有一个缺点，就是需要做乘法运算，这对于硬件插补器或者使用汇编语言的软件插补器带来了一定的困难。为了简化偏差值F_m的计算，通常采用迭代公式，即根据当前点的偏差值推算出下一点的偏差值。下面推导简化的偏差计算公式。

(1) 设加工点正处于m点，当$F_m \geq 0$时，应沿$+x$方向进一步至$(m+1)$点，该点的坐标值为：$x_{m+1} = x_m + 1$，$y_{m+1} = y_m$，则该点的偏差为：

$$F_{m+1} = y_{m+1} \cdot x_e - x_{m+1} \cdot y_e = y_m \cdot x_e - (x_m + 1) \cdot y_e = F_m - y_e$$

(2) 设加工点正处于m点，当$F_m < 0$时，表明m点在OA下方，应向$+y$方向进给一步至$(m+1)$点，该点的坐标值为：$x_{m+1} = x_m$，$y_{m+1} = y_m + 1$，则该点的偏差为：

$$F_{m+1} = y_{m+1} \cdot x_e - x_{m+1} \cdot y_e = (y_m + 1) \cdot x_e - x_m \cdot y_e = F_m + x_e$$

简化后偏差计算公式中只有一次加法或减法运算，新的加工点的偏差F_{m+1}都可以由前

一点偏差 F_m 和终点坐标相加或相减得到。需要注意的是，加工的起点是坐标原点，起点的偏差是已知的，即 $F_0=0$。

2）终点判断方法

用逐点比较法进行加工控制时，需随时确定刀具是否已经抵达直线终点。如果到了终点，则停止插补计算，否则继续循环处理插补计算。常用的终点判别方法有以下三种。

（1）总步长法

在插补处理开始之前，先设置一个总步长计数器 N_{xy}，其初值为：

$$N_{xy}=|x_e|+|y_e|$$

其中，$|x_e|$：在 X 轴方向上刀具应该走的总步数；

$|y_e|$：在 Y 轴方向上刀具应该走的总步数；

N_{xy}：整个插补过程中，刀具应该走的总步数。

在插补过程中，每进行一次插补计算，无论哪根坐标轴进给一步，计数器 N_{xy} 都做一次减 1 操作。当计数器 N_{xy} 内容减到零时，表示刀具已经走完了规定的步数，抵达直线轮廓的终点，则系统停止插补计算。

（2）终点坐标法

在插补处理开始之前，先设置两个步长计数器 N_x 和 N_y，分别用来存放刀具在两个坐标轴方向上应该走的总步数：$N_x=|x_e|$，$N_y=|y_e|$。

在插补过程中，每进行一次插补计算，如果 X 方向进给一步，则计数器 N_x 做减 1 操作；如果 Y 方向进给一步，则计数器 N_y 做减 1 操作。当两个步长计数器都为零时，表示刀具已经抵达直线轮廓的终点，系统停止插补计算。

（3）投影法

在插补处理开始之前，先确定直线轮廓终点坐标绝对值中较大的那根轴，并求出此轴方向应该运动的总步数，然后存放在总步长计数器 N 中：$N=\max(|x_e|,|y_e|)$。

在插补过程中，每进行一次插补计算，如果终点坐标绝对值较大的那根坐标轴进给一步，则计数器 N 做减 1 操作。当计数器 N 内容减到零时，表示刀具在终点坐标绝对值较大的那根坐标轴方向上已经走了规定的步数，应该已经抵达直线轮廓的终点，系统停止插补计算。

3）四个象限的直线插补

根据上述第一象限插补的基本过程，可以同理推导出其他三个象限过坐标原点的直线，从原点出发的插补算法，每个象限的偏差计算公式和进给方式如表 7.1 所示；直线插补时

表 7.1 过原点直线插补的进给方向和偏差计算公式

$F_m\geqslant 0$			$F_m<0$		
直线所在象限	进给方向	偏差计算公式	直线所在象限	进给方向	偏差计算公式
一、四	$+x$	$F_{m+1}=F_m-y_e$	一、二	$+y$	$F_{m+1}=F_m+x_e$
二、三	$-x$		三、四	$-y$	

偏差符号与进给方向关系图如图 7.7 所示。

4）直线插补举例

【例 7.1】 设加工第一象限一条直线 OA，起点为原点 $O(0,0)$，终点为 $A(4,5)$，试列表进行插补计算，并做出走步的轨迹图，同时给出算法程序流程图。

解：(1) 根据起点和终点坐标，以及驱动设备的步长，可确定步长计数器初值设为 $N_{xy}=|x_e|+|y_e|=4+5=9$，其中 $x_e=4$，$y_e=5$；初始偏差值 $F_0=0$；直线在第一象限，表示为 $xoy=1$。

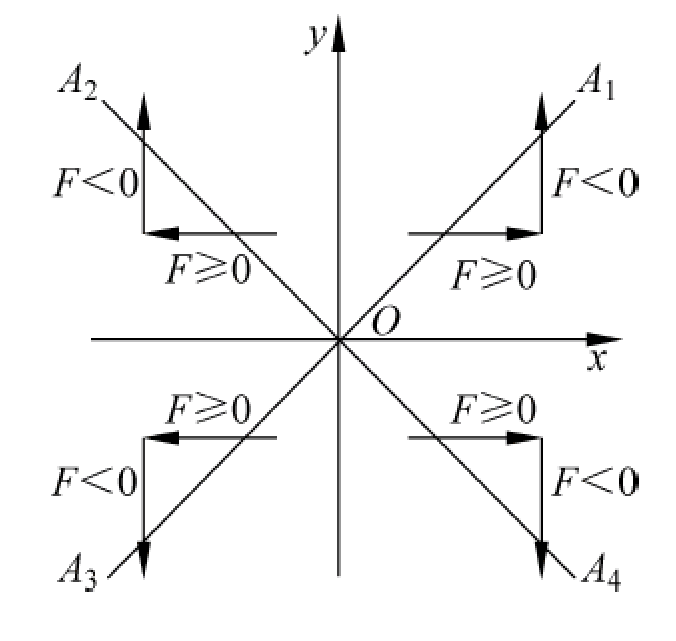


图 7.7　直线插补时偏差符号与进给方向关系图

插补计算过程如表 7.2 所示。

表 7.2　插补计算过程

步数	偏差判别	坐标进给	偏差计算	终点判断
起点			$F_0=0$	$N_{xy}=9$
1	$F_0=0$	$+x$	$F_1=F_0-y_e=0-5=-5$	$N_{xy}=8$
2	$F_1=-5<0$	$+y$	$F_2=F_1+x_e=-5+4=-1$	$N_{xy}=7$
3	$F_2=-1<0$	$+y$	$F_3=F_2+x_e=-1+4=3$	$N_{xy}=6$
4	$F_3=3>0$	$+x$	$F_4=F_3-y_e=3-5=-2$	$N_{xy}=5$
5	$F_4=-2<0$	$+y$	$F_5=F_4+x_e=-2+4=2$	$N_{xy}=4$
6	$F_5=2>0$	$+x$	$F_6=F_5-y_e=2-5=-3$	$N_{xy}=3$
7	$F_6=-3<0$	$+y$	$F_7=F_6+x_e=-3+4=1$	$N_{xy}=2$
8	$F_7=1>0$	$+x$	$F_8=F_7-y_e=1-5=-4$	$N_{xy}=1$
9	$F_8=-4<0$	$+y$	$F_9=F_8+x_e=-4+4=0$	$N_{xy}=0$

(2) 根据以上插补计算，可以画出直线插补的走步轨迹图如图 7.8 所示。

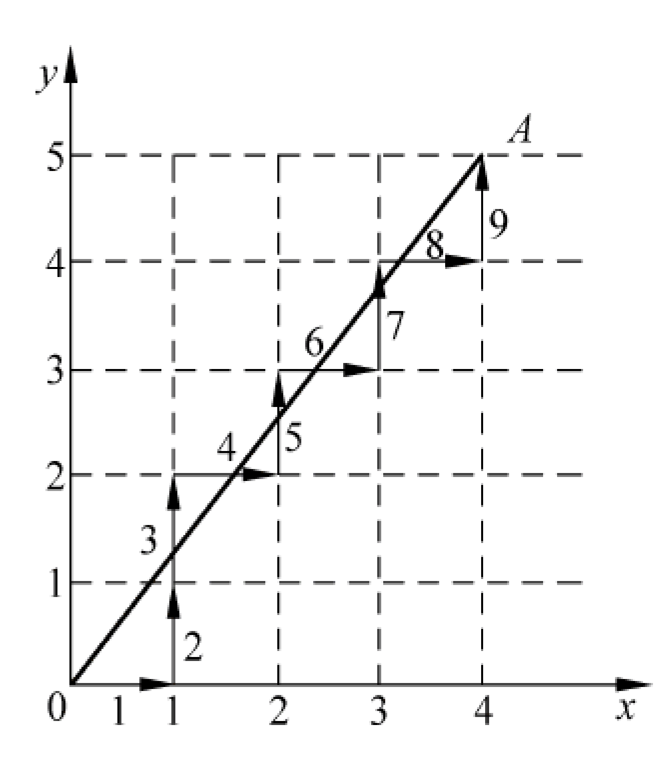


图 7.8　直线插补的走步轨迹图

(3) 根据直线插补的 4 个步骤：偏差判别、坐标进给、偏差计算、终点判断，下面做插补算法的程序流程图。其中，在计算机的内存中开辟 6 个单元。

XE、*YE*：存放终点横坐标 x_e、终点纵坐标 y_e。

NXY：存放总步数 N_{xy}。

FM：存放加工点偏差 F_m，其初值 $F_0=0$。

XOY：存放直线所在象限值 xoy，xoy 等于 1、2、3、4 分别代表第一、第二、第三、第四象限，可由终点坐标 (x_e, y_e) 的正、负符号来确定。

ZF：存放走步方向标志。$ZF=1$、2、3、4 分别代表走步方向为：$+x$、$-x$、$+y$、$-y$。

直线插补算法程序流程图如图 7.9 所示。

其中，图 7.9 中"坐标进给"通常是通过给步进电机发走步脉冲，由步进电机带动机床工作台或刀具产生进给运动。具体方式在本章后续 7.3.2 节中说明。另外，偏差计算公式中的 *XE*、*YE* 均为绝对值。

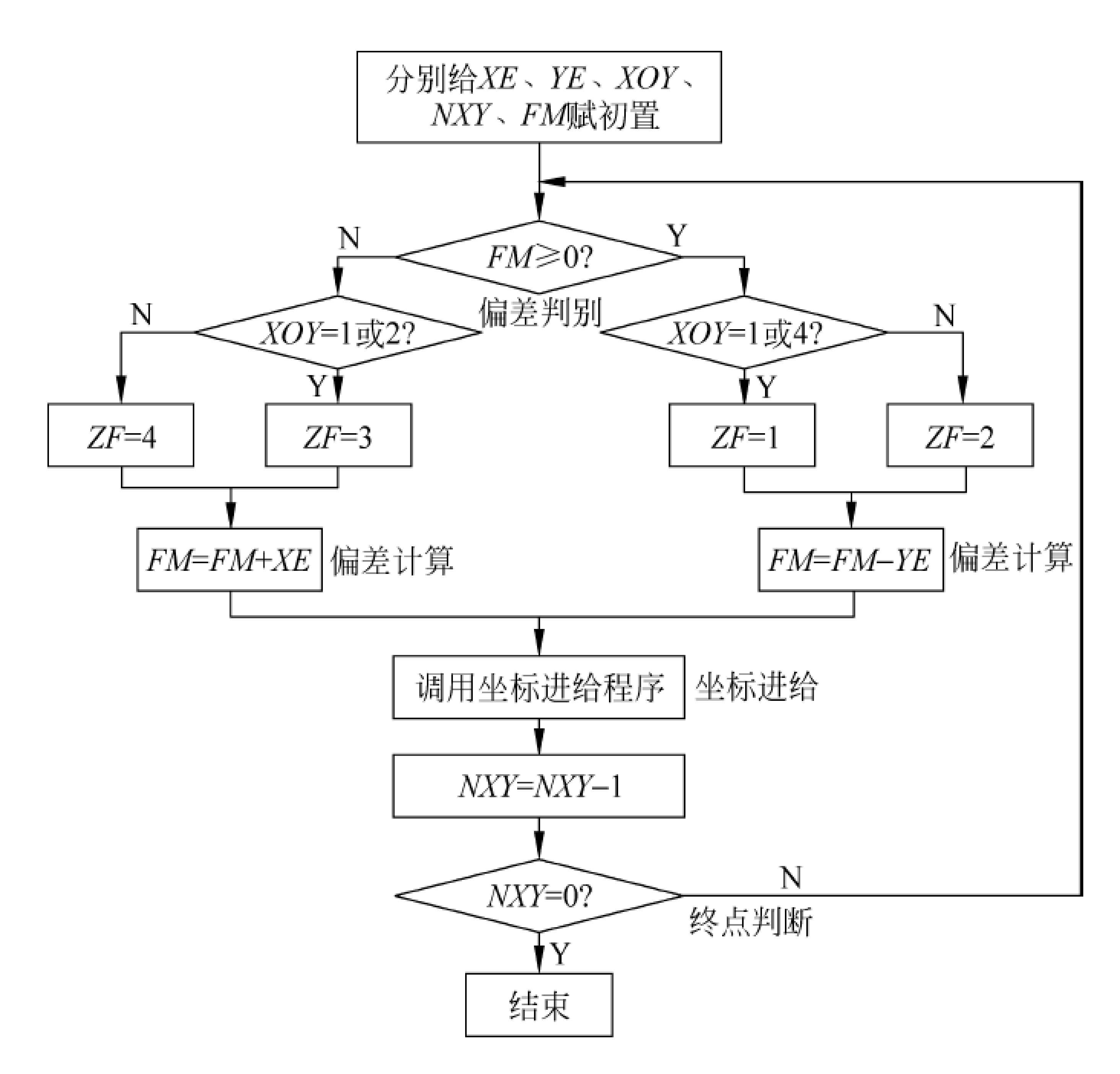


图 7.9　直线插补算法程序流程图

2. 逐点比较法圆弧插补

逐点比较法圆弧插补的基本方法是：数控系统的刀具或是工作台的某个运动轴每走一步就和给定轨迹上的坐标值进行一次比较，看这点在给定圆弧轨迹的里面还是外面，如果运动点在给定圆弧轨迹的里面，下一步就向给定轨迹的外面走，否则下一步向里面走，从而决定每一步的进给方向，如图 7.10 所示，即用折线来逼近给定轨迹。下面以图 7.11 所示第一象限逆圆弧 AB 的插补计算为例来说明圆弧插补的基本方法。

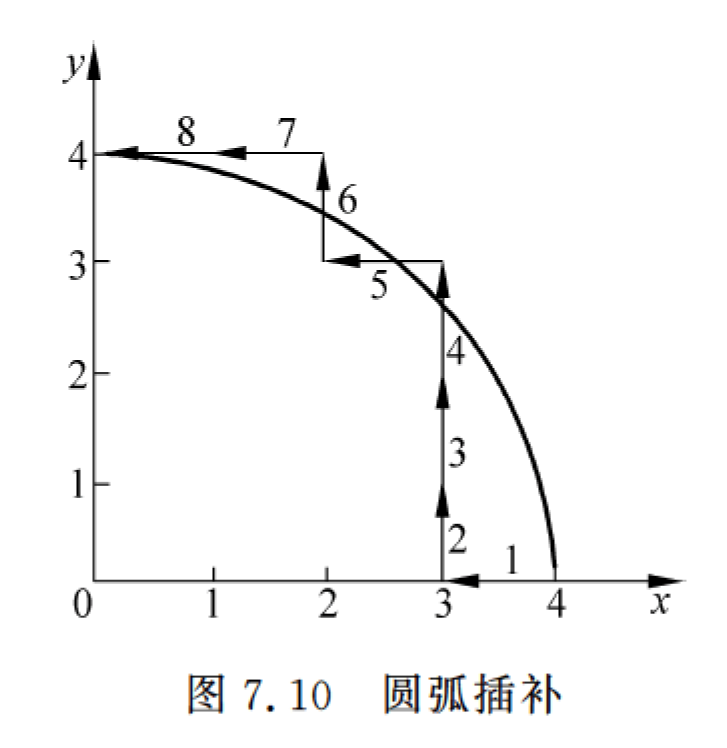


图 7.10　圆弧插补

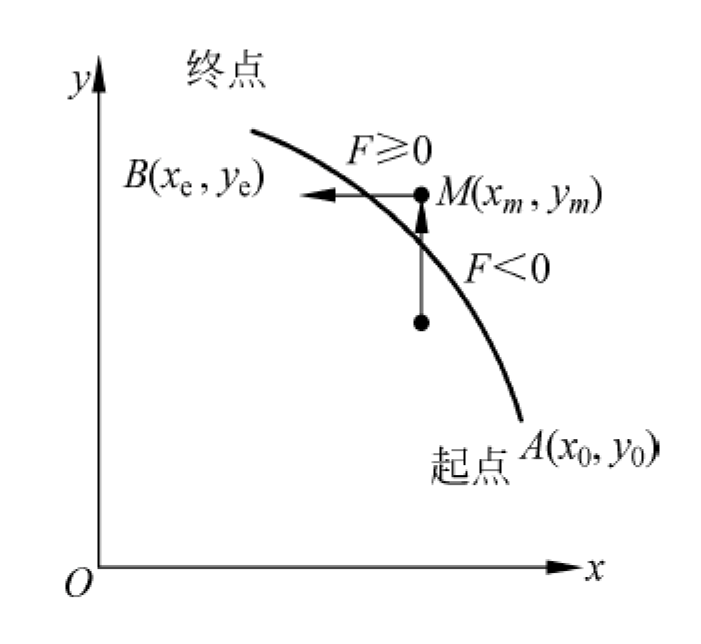


图 7.11　圆弧插补的位置偏差计算

1）位置偏差的计算

设要加工第一象限逆圆弧 AB，如图 7.11，圆弧的圆心在坐标原点，已知圆弧的起点为 $A(x_0, y_0)$，终点为 $B(x_e, y_e)$，圆弧半径为 R；进行圆弧加工时，设运动点为 $M(x_m, y_m)$。

由图所示的第一象限逆圆弧 AB 可知，点 M 到坐标原点的距离为 $R_m^2 = x_m^2 + y_m^2$，起点 A：$R^2 = x_0^2 + y_0^2$，可定义偏差判别式为：

$F_m=R_m^2-R^2=x_m^2+y_m^2-R^2$

$F_m=0$：表明加工点 M 在圆弧上。

$F_m>0$：表明加工点 M 在圆弧外。

$F_m<0$：表明加工点 M 在圆弧内。

因此，可以判断出直线插补的刀具进给方向为：从圆弧的起点出发，当 $F_m>0$，下一步向 $-x$ 方向进给一步；若 $F_m<0$，下一步向 $+y$ 方向进给一步；若 $F_m=0$，既可以向 $-x$ 方向也可以向 $+y$ 方向进给一步，在此约定取 $-x$ 方向。

如此一步一步计算和进给，并在到达终点后停止计算，就可插补出图 7.11 所示的第一象限逆圆弧 AB。但是，偏差的计算公式需要做乘方运算，对于硬件插补器或者使用汇编语言的软件插补器，这将产生一定的困难。为简化偏差值 F_m 的计算，通常采用迭代公式，即根据当前点的偏差值推算出下一点的偏差值。根据这个思想，对上述偏差值计算公式进行简化。

(1) 设加工点正处于点 $M(x_m,y_m)$，当 $F_m\geqslant 0$ 时，应沿 $-x$ 方向进给一步至点 m'，其坐标值为：$x_{m+1}=x_m-1$，$y_{m+1}=y_m$。

则新的加工点的偏差为：$F_{m+1}=x_{m+1}^2+y_{m+1}^2-R^2=(x_m-1)^2+y_m^2-R^2=x_m^2-2x_m+1+y_m^2-R^2=F_m-2x_m+1$。

(2) 设加工点正处于点 $M(x_m,y_m)$，当 $F_m<0$ 时，应沿 $+y$ 方向进给一步至点 M'，其坐标值为：$x_{m+1}=x_m$，$y_{m+1}=y_m+1$。

则新的加工点偏差为：$F_{m+1}=x_{m+1}^2+y_{m+1}^2-R^2=x_m^2+(y_m+1)^2-R^2=x_m^2+y_m^2+2y_m+1-R^2=F_m+2y_m+1$。

因此，只要知道前一点的偏差 F_m 和坐标值 (x_m,y_m)，就可求出新一点的偏差。起点的偏差 $F_0=0$。

2) 终点判断方法

圆弧插补的终点判断方法和直线插补类似，只是计算公式略有不同。常用的终点判别方法有以下三种。

(1) 总步长法

在插补处理开始之前，可以先设置一个总步长计数器 N_{xy}，其初值为：

$$N_{xy}=|x_e-x_0|+|y_e-y_0|$$

其中，

$|x_e-x_0|$：刀具在 x 轴方向上应该走的总步数。

$|y_e-y_0|$：刀具在 y 轴方向上应该走的总步数。

N_{xy}：整个插补过程中，刀具应该走的总步数。

在插补过程中，每进行一次插补计算，无论哪根坐标轴进给一步，计数器 N_{xy} 都做一次减 1 操作。当计数器 N_{xy} 内容减到零时，表示刀具已经走完了规定的步数，抵达圆弧轮廓的终点，则系统停止插补计算。

(2) 投影法

在插补处理开始之前，先确定所走步数较大的那根轴，并求出该轴运动的总步数，然后存放在总步长计数器 N 中。

$$N=\max(|x_e-x_0|,|y_e-y_0|)$$

在插补过程中，每进行一次插补计算，如果需走步数较大的那根坐标轴进给一步，则计数器 N 做一次减1操作，当计数器 N 内容减到零时，表示刀具在所走步数较大的那根坐标轴方向上已经走了规定的步数，应该已经抵达直线轮廓的终点，系统停止插补计算。

(3) 终点坐标法

先设置两个步长计数器 N_x 和 N_y，分别用来存放刀具在两个坐标轴方向上应该走的总步数：$N_x=|x_e-x_0|$，$N_y=|y_e-y_0|$。

在插补过程中，每进行一次插补计算，如果 x 方向进给一步，则计数器 N_x 做一次减1操作；如果 y 方向进给一步，则计数器 N_y 做一次减1操作。当两个步长计数器都为零时，表示刀具已经抵达直线轮廓的终点，系统停止插补计算。

3) 圆弧插补计算过程

由于圆弧插补计算过程中，每一步偏差 F_{m+1} 的计算不仅需要 F_m，还需要运动点的坐标 x_m 或 y_m，因此，插补过程比直线插补计算过程多一个环节，即要计算加工点瞬时坐标（动点坐标）值。这样，圆弧插补计算过程分为5个步骤，即偏差判别、坐标进给、偏差计算、坐标计算、终点判断。

4) 四个象限的圆弧插补

根据上述第一象限逆时针圆弧插补的基本过程，可以同理推导出其他三个象限的圆弧插补方法（顺圆弧和逆圆弧），每个象限的偏差计算公式和进给方式如表7.3所示；圆弧插补时偏差符号与进给方向关系如图7.12所示。其中 CA 表示顺圆弧（Clockwise Arc），$CA1$、$CA2$、$CA3$、$CA4$ 分别表示第一、二、三、四象限的顺圆弧；IA 表示逆圆弧（Inverse Arc），$IA1$、$IA2$、$IA3$、$IA4$ 分别表示第一、二、三、四象限的逆圆弧。

表7.3 四个象限的圆弧插补偏差计算公式和进给方式

位置偏差	圆弧种类	进给方向	偏差计算公式	坐标计算
$F_m\geqslant 0$	$CA1$、$IA2$	$-y$	$F_{m+1}=F_m-2y_m+1$	$x_{m+1}=x_m$ $y_{m+1}=y_m-1$
	$CA3$、$IA4$	$+y$		
	$CA4$、$IA1$	$-x$	$F_{m+1}=F_m-2x_m+1$	$x_{m+1}=x_m-1$ $y_{m+1}=y_m$
	$CA2$、$IA3$	$+x$		
$F_m<0$	$CA1$、$IA4$	$+x$	$F_{m+1}=F_m+2x_m+1$	$x_{m+1}=x_m+1$ $y_{m+1}=y_m$
	$CA3$、$IA2$	$-x$		
	$CA2$、$IA1$	$+y$	$F_{m+1}=F_m+2y_m+1$	$x_{m+1}=x_m$ $y_{m+1}=y_m+1$
	$CA4$、$IA3$	$-y$		

5) 圆弧插补举例

【例7.2】 设加工第一象限一条顺圆弧 AB，起点为点 $A(4,3)$，终点为 $B(5,0)$，试列表进行插补计算，并做出走步的轨迹图，同时给出算法程序流程图。

解：(1) 根据起点和终点坐标，以及驱动设备的步长，可确定步长计数器初值设为 $N_{xy}=|x_e-x_0|+|y_e-y_0|=|5-4|+|0-3|=1+3=4$，其中 $x_e=5$，$y_e=0$；$x_0=4$，$y_0=3$；初始偏差值 $F_0=0$。

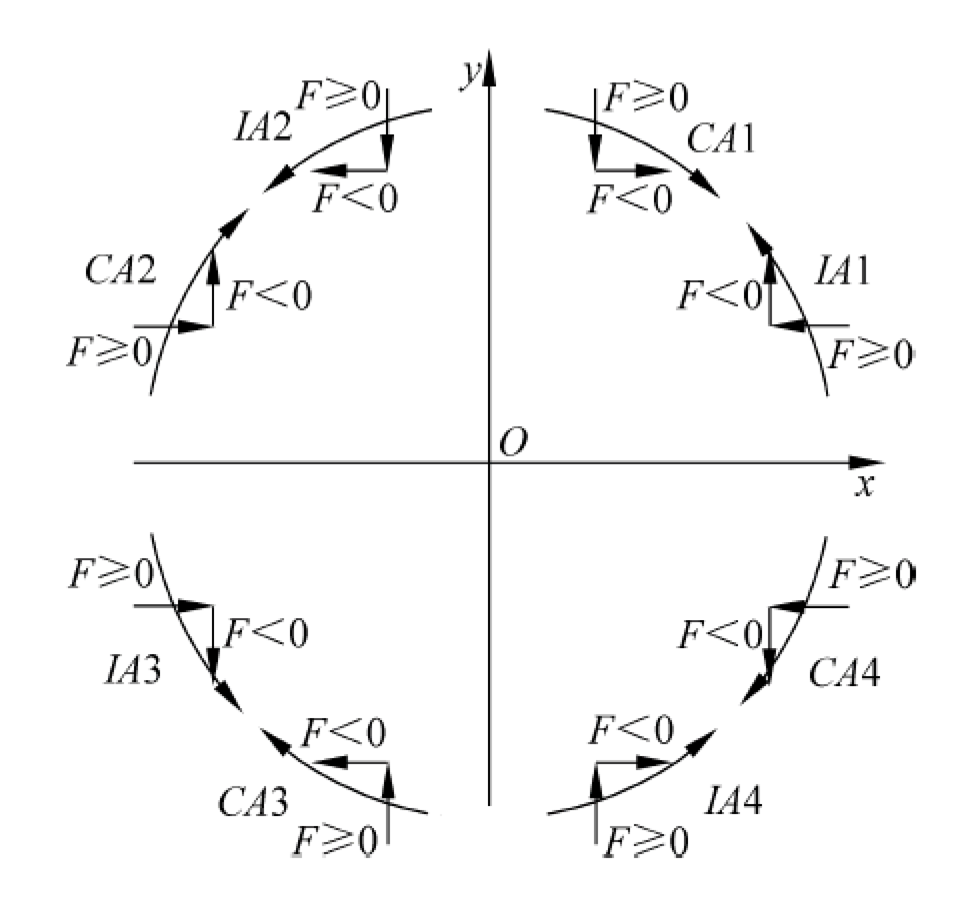


图 7.12　圆弧插补时偏差符号与进给方向关系图

根据表 7.3 中所示第一象限顺圆弧（CA1）的偏差与进给方向、坐标计算的关系，得到插补计算过程如表 7.4 所示。

表 7.4　圆弧插补计算过程

步数	偏差判别	进给方向	偏差计算	坐标计算	终点判断
起点			$F_0=0$	$x_0=4, y_0=3$	$N_{xy}=4$
1	$F_0=0$	$-y$	$F_1=F_0-2y_0+1=-5$	$x_1=x_0=4$ $y_1=y_0-1=3-1=2$	$N_{xy}=3$
2	$F_1=-5<0$	$+x$	$F_2=F_1+2x_1+1=4$	$x_2=x_1+1=5$ $y_2=y_1=2$	$N_{xy}=2$
3	$F_2=4>0$	$-y$	$F_3=F_2-2y_2+1=1$	$x_3=x_2=5$ $y_3=y_2-1=2-1=1$	$N_{xy}=1$
4	$F_3=1>0$	$-y$	$F_4=F_3-2y_3+1=0$	$x_4=x_3=5$ $y_4=y_3-1=1-1=0$	$N_{xy}=0$

（2）根据以上插补计算过程，可以画出此圆弧插补的走步轨迹图如图 7.13 所示。

（3）根据圆弧插补的 5 个步骤：偏差判别、坐标进给、偏差计算、坐标计算、终点判断，下面做插补算法的程序流程图。其中，在计算机的内存中开辟八个单元。

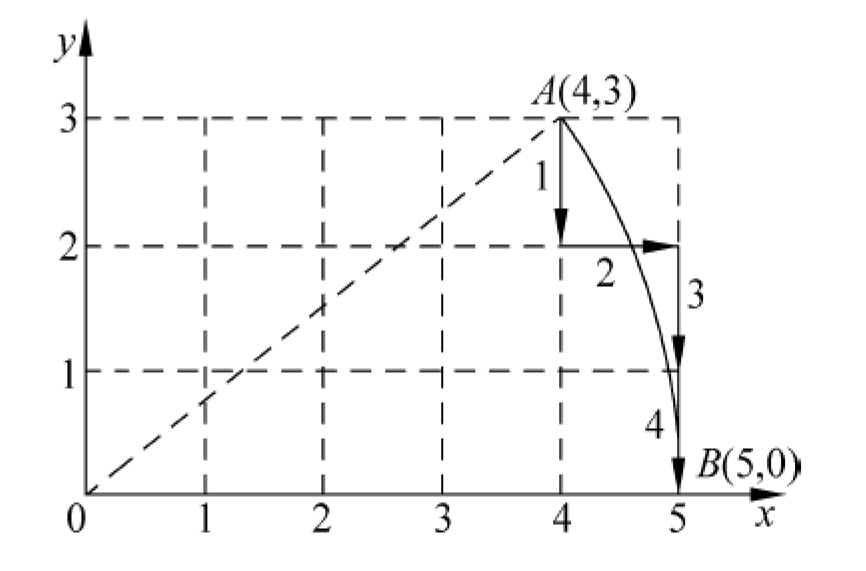


图 7.13　圆弧插补的走步轨迹图

X0、Y0：存放起点横坐标 x_0、起点纵坐标 y_0。

NXY：存放总步数 N_{xy}。

FM：存放加工点偏差 F_m，其初值 $F_0=0$。

RCI：存放圆弧种类值，RCI 等于 1、2、3、4 和 5、6、7、8 分别代表顺圆弧 CA1、CA2、CA3、CA4 和逆圆弧 IA1、IA2、IA3、IA4。RCI 的值可由起点和终点的坐标的正、负符号来确定。

XM、YM：存放加工运动点横坐标 x_m 和纵坐标 y_m，x_m 和 y_m 的初值为 x_0 和 y_0。

ZF：存放走步方向标志，ZF=1、2、3、4 分别代表走步方向为：$+x$、$-x$、$+y$、$-y$。

此圆弧插补算法程序流程图如图 7.14 所示。

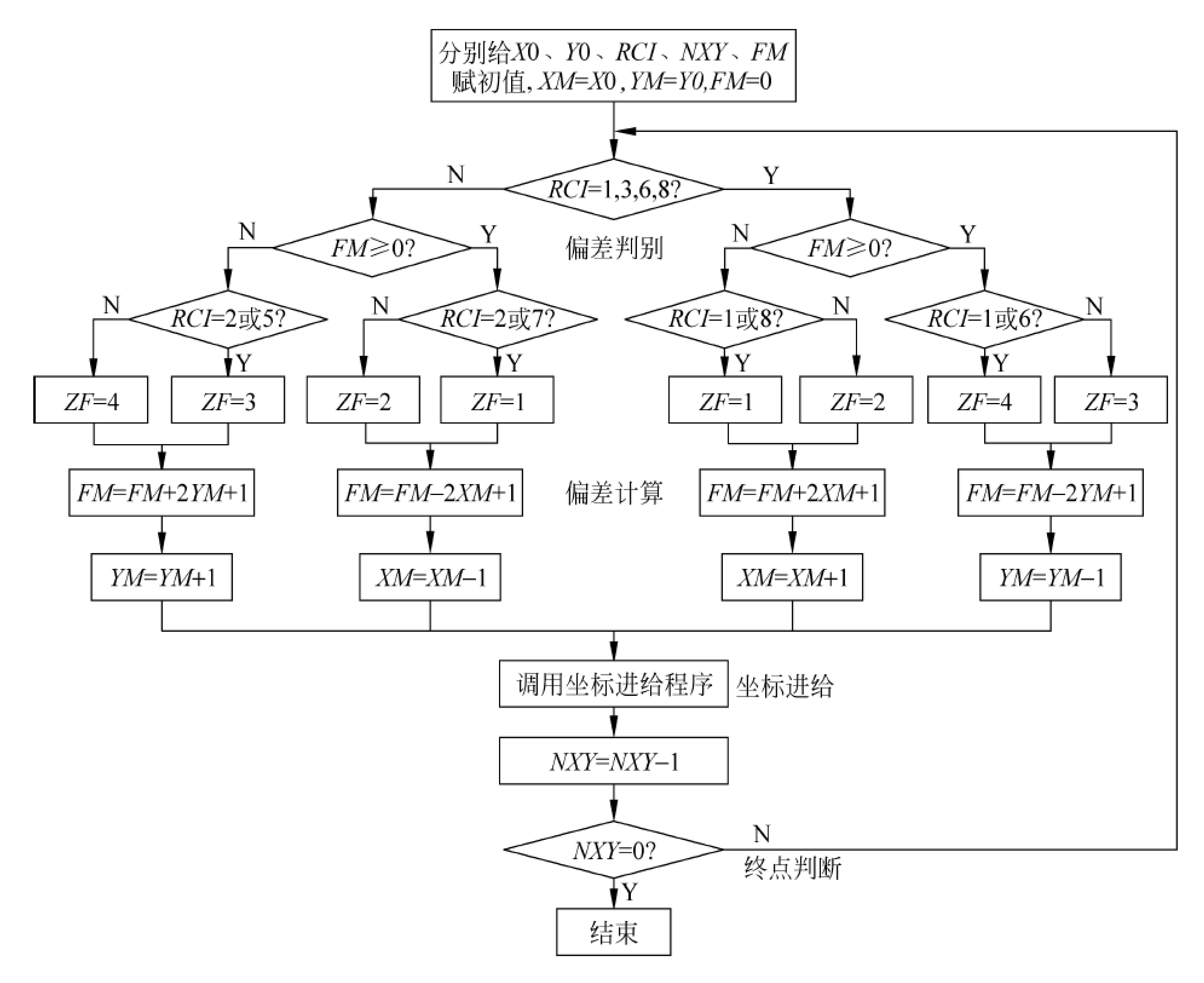


图 7.14 圆弧插补算法程序流程图

其中,图 7.14 中"坐标进给"通常也是通过给步进电机发走步脉冲,由步进电机带动机床工作台或刀具产生进给运动。同时,偏差计算公式中的 XM、YM 均为绝对值。

7.3 步进驱动数字程序控制技术

步进电机是工业控制的主要驱动装置之一,它不仅可以直接接收计算机输出的数字信号,不需要进行数/模转换,而且其具有快速启停、角位移与控制脉冲间精确同步的特点,可以用于开环控制系统。只需利用适当的机械结构将角位移转变为位置、体积、流量等物理量的变化,便可实现对相应的工业对象进行控制。所以步进电机被广泛地应用于数字控制系统中。如在数控机床等加工设备中,可以利用丝杠把步进电机的角度变化变成直线位移,实现机床工作平台的定位控制。又如绘图机、打印机及光学仪器中,采用步进电动机来定位绘图笔、印字头或光学镜头等。步进电机在工业过程控制的位置控制系统中的应用越来越广泛。

7.3.1 步进电机的结构及工作原理

步进电机又称脉冲电机,是靠脉冲来驱动的,给一个脉冲信号电机转一拍,是一种将电

脉冲信号转换为角位移的机电式数模(D/A)转换器。其主要特点如下。

- 总移动步数决定于指令脉冲的总数。
- 移动的速度则取决于指令脉冲的频率,但不能超过最高频率,否则电机的力矩减小,电机不能正常转动。
- 移动的方向则取决于指令脉冲的相序。

1. 步进电机的分类

不同类型步进电机其工作原理、驱动装置也不完全一样,但其工作过程基本是相同的。下面从不同角度讨论步进电机的分类。

(1) 步进电机按运动方式来分,可分为旋转运动式、直线运动式、平面运动式(印刷绕组式)和滚切运动式。

(2) 按工作原理来分,可分为反应式(磁阻式)、电磁式、永磁式、永磁感应式(混合式)步进电机。

(3) 按其工作方式来分,分为功率式和伺服式。前者输出转矩较大,能直接带动较大的负载;后者输出转矩较小,只能带动较小的负载,但精确度较高;对于大负载需通过液压放大元件来传动。

(4) 按结构来分,可分为单段式(径向式)、多段式(轴向式)和印刷绕组式。

(5) 按相数来分,可分为三相步进电机、四相步进电机、五相步进电机、六相步进电机等。

(6) 按使用频率来分,可分为高频步进电机和低频步进电机。

2. 三相反应式步进电机的基本结构

步进电动机和一般旋转电动机一样,分为定子和转子两大部分,下面以三相反应式步进电机为例介绍步进电机的基本结构,如图 7.15 所示。

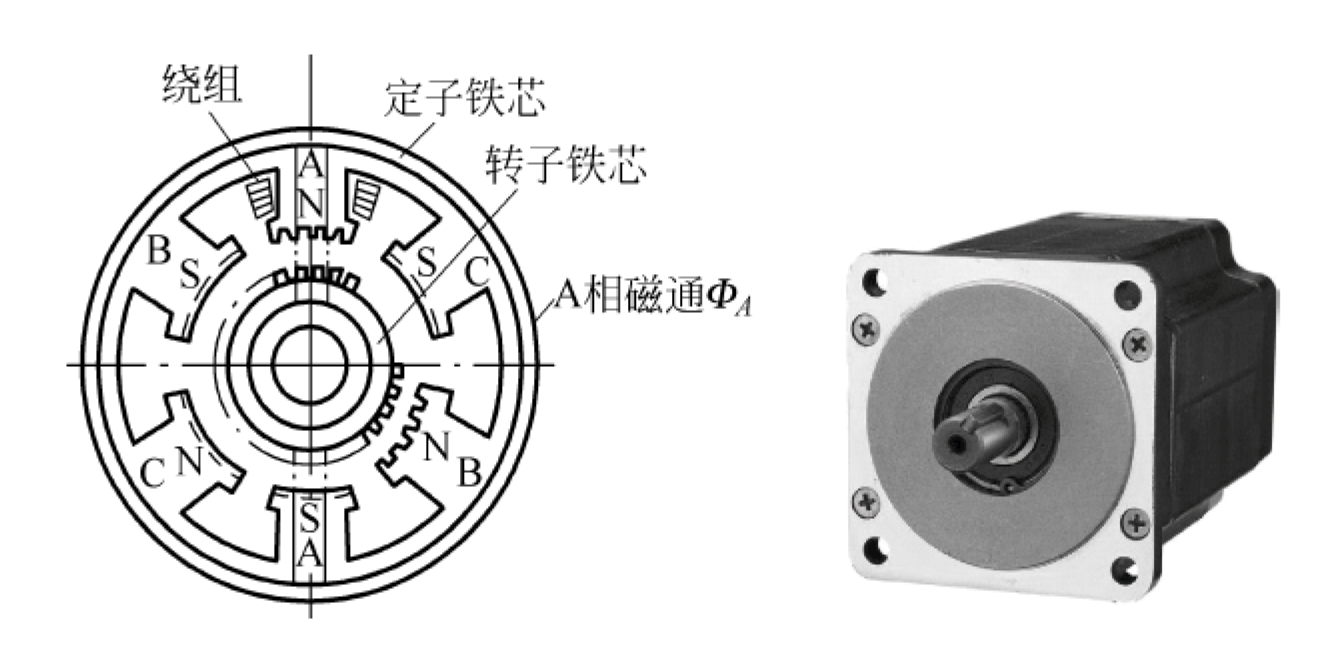


图 7.15　三相反应式步进电机的基本结构与外形

步进电机定子铁芯由硅钢片叠压而成,定子上有 6 个均匀分布的磁极,每两个为一对。定子绕组绕置在定子 6 个均匀分布的铁芯上,把沿直径方向上相对的两个磁极上的线圈串联在一起,构成一相控制绕组。图 7.15 中所示的步进电机有 A、B、C 三相控制绕组,故称为三相步进电机。当任意一相绕组通电时,便形成一对定子磁极,即形成 N、S 极。

转子用硅钢片叠成或用软磁性材料做成凸极结构。反应式步进电动机的转子本身没有

励磁绕组,用永久磁铁做转子的称为永磁式步进电动机。目前我国以反应式步进电动机用得较多。

在定子的每个磁极上,即定子铁芯的每个齿上又开了五个小齿,齿槽等宽,齿间夹角为9°,在空间位置上依次错开1/3齿距;转子是一个带齿的铁芯,没有绕组,转子上均匀分布了40个小齿(图7.15所示三相反应式步进电机),齿槽等宽,齿间夹角也是9°(360°/40=9°),与定子磁极上的小齿一致。定子齿与转子齿展开图如图7.16所示,当定子A相磁极上的小齿与转子上的小齿对齐时,定子B相磁极上的小齿刚好超前(或滞后)转子小齿1/3齿距角(3°),定子C相磁极上的小齿超前(或滞后)转子小齿2/3齿距角(6°)。

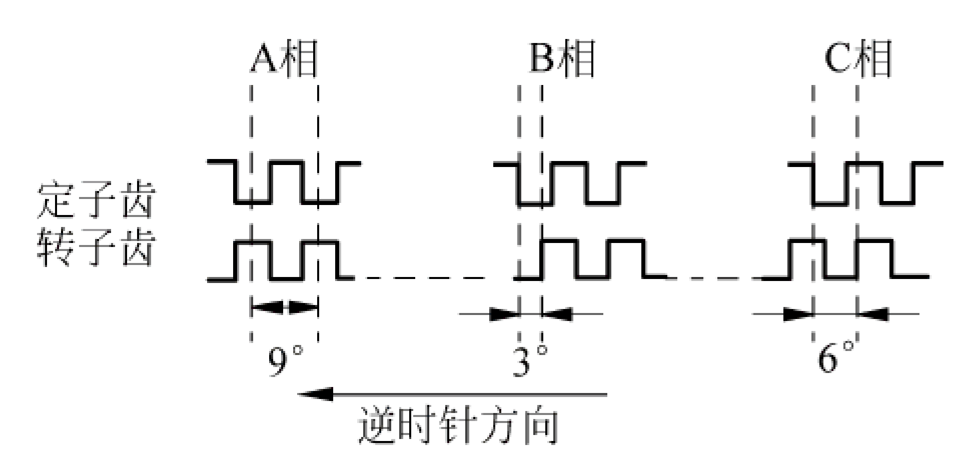


图7.16 定子齿与转子齿展开图

3. 三相反应式步进电机的工作原理

步进电动机的工作原理是基于电磁感应,是利用电磁铁的原理,磁力线总是力图走磁阻最小的路径,从而产生反应力矩,将脉冲信号转换成线位移或角位移。下面仍以图7.16所示三相反应式步进电机为例简述其工作原理。

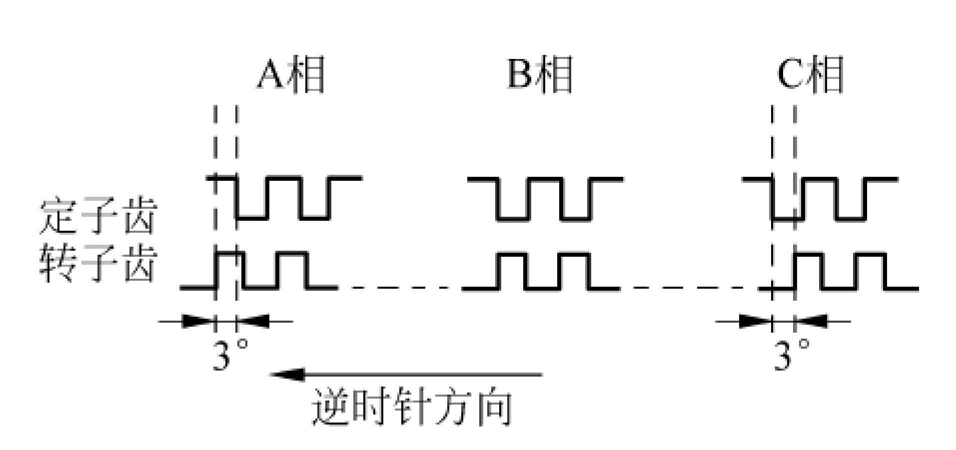
(1) 当A相通电,B、C都不通电,将转子齿吸引到定子磁极A相下,并与A相齿对齐,如图7.16所示。因转子只受到径向力而无切向力,故转矩为零,转子被自锁在这个位置。此时,定子B相与转子齿错开1/3齿距,定子C相与转子齿错开2/3齿距。我们把这个状态当做一个初状态。

(2) 当B相绕组通电,A、C相不通电,仍然因磁通总是要沿着磁阻最少的路径闭合,将使转子齿和定子磁极B的齿相对齐,在电磁力矩的作用下使转子沿逆时针方向转过3°(1/3齿距),如图7.17(a)所示。此时,定子A相滞后转子齿1/3齿距,定子C相超前转子齿2/3齿距。

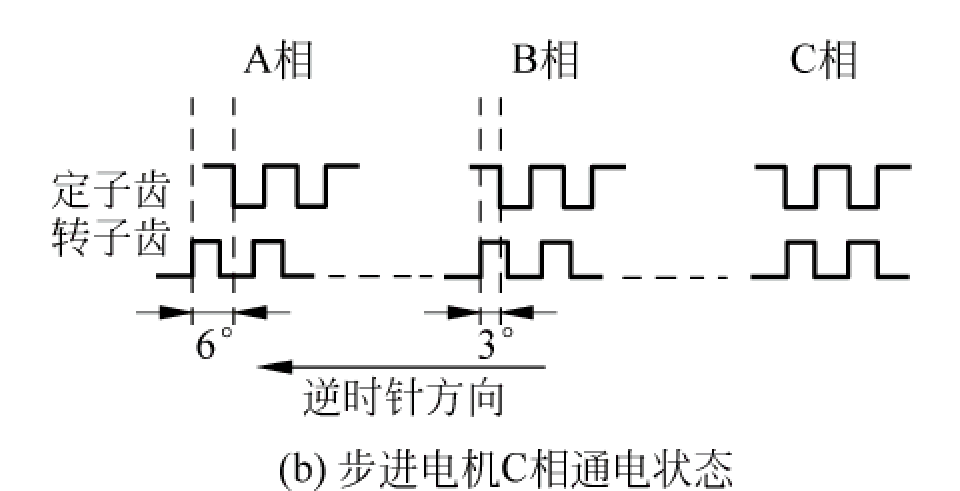
(3) 当C相通电,A、B相断电,又使转子齿和定子磁极C对齐,在电磁力矩的作用下使转子再沿着逆时针方向转过3°,如图7.17(b)所示。此时,定子A相滞后转子齿2/3齿距,定子B相滞后转子齿1/3齿距。

(4) 当A相再次通电,B、C相断电,使转子齿和定子磁极A对齐,在电磁力矩的作用下使转子再沿着逆时针方向转过3°,再次回到定子A相与转子齿对齐,定子B相与转子齿错开1/3齿距,定子C相与转子齿错开2/3齿距。

从一相通电改换成另一相通电,即通电方式改变一次叫“一拍”。那对A、B、C三相绕组轮流通电一次需要三拍,此三拍为一个周期。从上面分析看出,电机刚好接收一个周期的信号,经三拍操作,向逆时针方向转动一个齿距,即9°。由于按A→B→C→A相轮流通电,则磁场沿A、B、



(a) 步进电机B相通电状态



(b) 步进电机C相通电状态

图7.17 步进电机工作原理分析

C 方向转动了 360°空间角，而这时转子沿 ABC 方向转动了一个齿距的位置。

若使定子绕组按 A→B→C→A 的顺序通电，则步进电机转子便不停地沿逆时针方向转动。如果定子绕组的通电顺序改为 A→C→B→A，则步进电机转子沿顺时针方向转动。

4. 三相反应式步进电机的工作方式

步进电动机有单相轮流通电、双相轮流通电和单双相轮流通电的方式。前面所述的通电方式叫做“三相单三拍”的方式。如果通电相序为 AB→BC→CA→AB，则称为“三相双三拍”；通电相序若为 A→AB→B→BC→C→CA→A，称为“三相单双六拍”。下面对此三种方式做一比较。

(1) 从一相绕组的通电或断电切换到另一相绕组的断电或通电时，由于电机绕组是电感性元件，磁场的消失或建立均需一定时间，因此，切换期间容易使电机产生失步。此外，由单一绕组通电吸引转子，也容易使转子在平衡位置附近产生振荡，运行的稳定性较差，所以很少采用。

(2) 三相双三拍工作方式：为了克服三相单三拍工作方式的不足，通常将其改为三相双三拍通电方式，即按 AB→BC→CA→AB 方式通电，可使电机正转；反之按 BA→AC→CB→BA 方式通电，可使电机反转。这种工作方式每个通电状态均为两相绕组同时通电，且相邻的两个工作节拍中总有一相保持通电的状态不变，因此，可避免三相单三拍工作方式的失步和低频振荡等现象。三相双三拍工作方式下，每改变一次通电状态可使图 7.16 所示步进电机旋转 3°。

(3) 三相单双六拍工作方式：为减小每改变一次通电状态使步进电机所转过的角度，而采用三相单双六拍通电方式。即按 A→AB→B→BC→C→CA→A 方式通电，使电机正转；反之，按 A→AC→C→CB→B→BA→A 方式通电，使电机反转。三相六拍工作方式除具有三相双三拍的特点外，还可使步进电机每改变一次通电状态旋转角度只有三相双三拍工作方式的一半，即 1.5°。

5. 步进电机特性计算公式

以如图 7.15 所示三相反应式步进电动机为例，下面介绍一下相关特性计算公式。

(1) 步进电机步距角的计算

设转子的齿数为 z，则齿距角 t_b 计算公式如下：

$$t_b = \frac{360^\circ}{z} \tag{7.1}$$

因为每改变一次通电状态(即运行一拍)，转子就走一步；各相绕组轮流通电一次，转子就转过一个齿距。故步距角计算公式如下：

$$\theta_b = \frac{\text{齿距}}{\text{相数}} = \frac{\text{齿距}}{Km} = \frac{360^\circ}{Kmz} \tag{7.2}$$

其中，m 为相数；K 为定子绕组通电方式系数。若相邻两次通电的相数一样，取 $K=1$，如三相单三拍、三相双三拍工作方式；反之，取 $K=2$，如三相单双六拍工作方式。

因此，7.3.1 节第 3 小节中，步进电动机的转子齿数 $z=40$，按三相单三拍运行时，$K=1$，$m=3$，所以步距角 $\theta_b=360^\circ/(1\times3\times40)=3^\circ$。若按三相单双六拍运行时，则 $K=2$，步距

角 $\theta_b=360°/(2\times3\times40)=1.5°$。

由以上可见，步进电动机的相数和转子齿数越多，步距角就越小，走步控制就越精确。可根据具体工程需要进行选择。

(2) 步进电机转速的计算

大家了解到步进电机的速度取决于控制脉冲的频率。设步进电动机通电脉冲频率为 f（脉冲数/秒），步距角 θ_b 用弧度表示，则步进电动机的转速计算公式可以表示为：

$$n=\frac{\theta_b}{2\pi}\cdot f\cdot 60=\frac{60f\dfrac{2\pi}{Kmz}}{2\pi}=\frac{60f}{Kmz}(\mathrm{r/min}) \tag{7.3}$$

因此，步进电动机在一定频率的控制脉冲作用下，电动机的相数和转子齿数越多，转速就越低；当电动机的相数和转子齿数一定的情况下，控制脉冲频率越高，步进电机的速度就越高。当然，步进电机相数越多，驱动电源也越复杂。

(3) 步进电机的脉冲当量

每一个控制脉冲控制机床运动部件产生的相应位移量称为脉冲当量，又称做最小设定单位。当选用步进电动机作为驱动设备时，一般是通过减速器和丝杠带动工作台进行移动，所以其步距角 θ_b 对应工作台的移动量便是步进系统工作台的最小运动单位，即脉冲当量 δ，单位为：mm/脉冲。其计算公式为：

$$\delta=\frac{\theta_b t}{360i} \tag{7.4}$$

其中，t 为丝杠导程(mm)，就是丝杠旋转一周带来的直线行程，也有的叫螺距；i 为减速装置的传动比。

这样可以得到工作台的进给速度 v(mm/min)的计算公式为：

$$v=60\delta f \tag{7.5}$$

【例 7.3】 一台五相步进电动机，采用五相十拍脉冲分配方式，转子有 100 个齿，脉冲频率为 $f=1000\text{Hz}$，丝杠导程为 $t=10\text{mm}$，减速装置的传动比 $i=10$。求：(1)写出一个循环的通电顺序；(2)求步进电动机步距角；(3)求步进电动机转速；(4)求步进电机的脉冲当量。

解：(1) 电机正转时，一个循环的通电顺序为：

A→AB→B→BC→C→CD→D→DE→E→EA→A

电机反转时，一个循环通电顺序为：

A→AE→E→ED→D→DC→C→CB→B→BA→A

(2) 根据步距角的计算公式：

$$\theta_b=\frac{\text{齿距}}{\text{拍数}}=\frac{\text{齿距}}{Km}=\frac{360°}{Kmz}$$

其中，$K=2$；$m=5$；$z=100$，因此有：

$$\theta_b=\frac{360°}{Kmz}=\frac{360°}{2\times5\times100}=0.36°$$

(3) 根据步进电机的速度计算公式：

$$n=\frac{\theta_b}{2\pi}\cdot f\cdot 60=\frac{60f\dfrac{2\pi}{Kmz}}{2\pi}=\frac{60f}{Kmz}(\mathrm{r/min})$$

其中，$f=1000\text{Hz}$；$K=2$；$m=5$；$z=100$，因此有：

$$n=\frac{60f}{Kmz}=\frac{60\times1000}{2\times5\times100}=60(\text{r/min})$$

（4）根据步进电机的脉冲当量计算公式：

$$\delta=\frac{\theta_b t}{360i}=\frac{360}{Kmz}\cdot\frac{t}{360i}$$

其中，$t=10\text{mm}$，$K=2$，$m=5$，$z=100$，$i=10$，因此有：

$$\delta=\frac{360}{Kmz}\cdot\frac{t}{360i}=\frac{360}{2\times5\times100}\cdot\frac{10}{360\times10}=0.001$$

7.3.2　步进电机的控制

步进电机常采用开环控制方式，其控制电路主要由控制器、脉冲分配器、驱动电路和步进电机等组成，如图 7.18 所示。

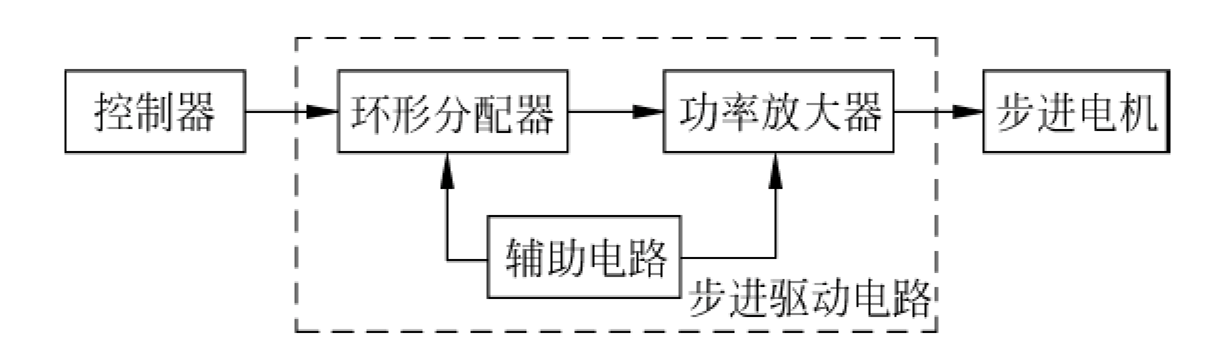


图 7.18　典型步进电机控制原理图

在 4.4.2 节“图表仿真方式”介绍中曾列举了一个步进电机控制系统的例子，并详细介绍了其仿真模型建立的方法。这里将其原理图重新给出，如图 7.19 所示，并以此系统为例，进一步分析步进电机的控制。其中，将 L297 输出的四相信号分别改为：A、B、C、D，L298 输出的四相信号改为：1A、1B、2A、2B。

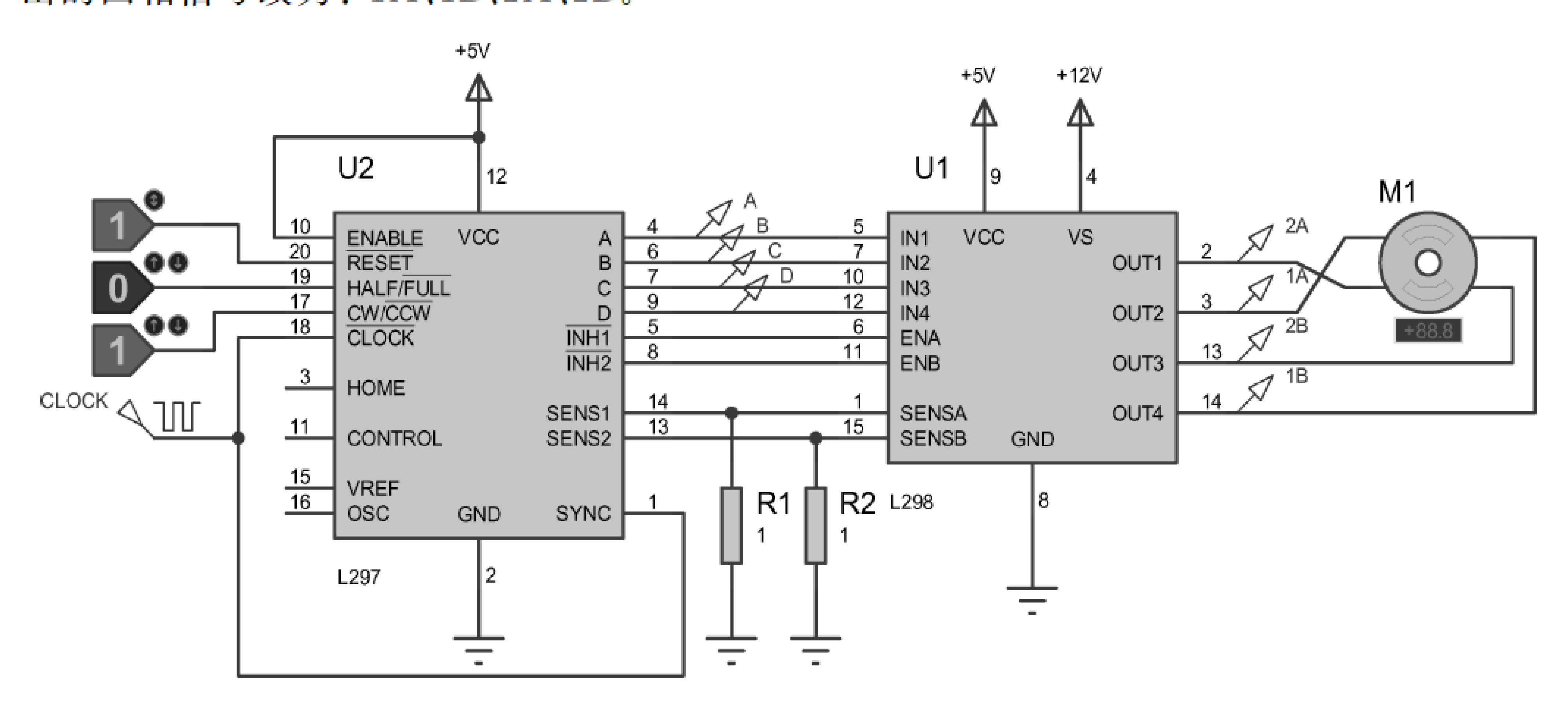


图 7.19　步进电机控制实例

1. 系统组成

图 7.19 所示实例搭建了一个典型的步进电机控制系统。其中 L297 是意大利 SGS 半导体公司生产的步进电机专用控制器。适用于双极性两相步进电机或单极性四相步进电机

的控制。它使用 5V 电源，全部信号的连接都与 TTL/CMOS 或集电极开路的晶体管兼容。其输出信号可控制 L298 双桥驱动集成电路，用来驱动电压最高为 46V，总电流为 4A 以下的步进电机；也可用来控制由达林顿管组成的分立电路，以驱动更高电压，更大电流的步进电机。L297 主要需要时钟、方向和模式输入信号，相位由内部产生，从而减轻了微处理器和程序设计的负担。L297 内部主要由译码器、两个固定斩波频率的 PWM 恒流斩波器以及输出控制逻辑组成，能够产生各种所需的相序。

L297 主要控制信号如下。

(1) SYNC(引脚 1)：斩波器输出端。如多个 L297 同步控制，所有的 SYNC 端都要连在一起，共用一套振荡元件。如果使用外部时钟源，则时钟信号接到此引脚上。

(2) GND(引脚 2)：接地端。

(3) HOME(引脚 3)：集电极开路输出端，当 L297 在初始状态(ABCD＝0101)时，此端有指示。当此引脚有效时，晶体管开路。

(4) A(引脚 4)：A 相驱动信号。

(5) $\overline{\text{INH1}}$(引脚 5)：控制 A 相和 B 相的驱动级，此引脚为低电平时，A 相和 B 相的驱动被禁止；当线圈绕组断电时，双极性桥用这个信号使负载电流快速衰减。若 CONTROL 端输入是低电平时，用斩波器调节负载电流。

(6) B(引脚 6)：B 相驱动信号。

(7) C(引脚 7)：C 相驱动信号。

(8) $\overline{\text{INH2}}$(引脚 8)：控制 C 相和 D 相的驱动级。作用与$\overline{\text{INH1}}$类似。

(9) D(引脚 9)：D 相驱动信号。

(10) ENABLE(引脚 10)：使能端，高电平有效。低电平时，$\overline{\text{INH1}}$、$\overline{\text{INH2}}$、A、B、C、D 都为低电平。当系统被复位时，用来阻止电机驱动。

(11) CONTROL(引脚 11)：斩波器功能控制端。低电平时使$\overline{\text{INH1}}$、$\overline{\text{INH2}}$起作用，高电平时，使 ABCD 输出端起作用。

(12) VCC(引脚 12)：＋5V 电源输入端。

(13) SENS2(引脚 13)：C 相和 D 相的绕组电流检测电压反馈输入端。

(14) SENS1(引脚 14)：A 相和 B 相的绕组电流检测电压反馈输入端。

(15) VREF(引脚 15)：斩波器基准电压输入端。加到此引脚的电压决定绕组电流的峰值。

(16) OSC(引脚 16)：斩波频率输入端，一个 RC 网络接至此引脚以决定斩波频率。在多个 L297 同步工作时，其中一个接到 RC 网络，其余的此引脚接地。

(17) CW/$\overline{\text{CCW}}$(引脚 17)：电机转动方向的控制端，控制电机顺时针转动，或逆时针转动。

(18) $\overline{\text{CLOCK}}$(引脚 18)：时钟信号输入端。

(19) HALF/$\overline{\text{FULL}}$(引脚 19)：进行电机的步距角选择，可选半步方式或整步方式。此引脚输入高电平时，为半步方式(四相八拍)；低电平时，为整步方式。如选择整步方式，变换器在奇数状态时，将得到两相整步相序(四相双四拍)，变换器在偶数状态时，将得到单相整步相序(四相单四拍)。

(20) $\overline{\text{RESET}}$(引脚 20)：复位输入端。

时钟信号取自时钟激励源，$\overline{\text{RESET}}$、HALF/$\overline{\text{FULL}}$和 CW/$\overline{\text{CCW}}$信号可由控制器产生，仿真时用逻辑状态输入信号来代替。

L298 是用来驱动步进电机的集成电路，采用双全桥式驱动器，其设计是为接受标准 TTL 逻辑电平信号和驱动电感负载的，如继电器、直流电动机、步进电动机等。

步进电机采用一个双极性两相步进电机，如图 7.20 所示。双极性步进电机与单极性步进电机相比，除了相对磁极的绕组是以并行方式接入电源，本质上是一样的。由于该例中 L298 对步进电机是双极性驱动，步进电机的定子励磁绕组线圈可以完全利用，从而使步进电机达到最佳的驱动状态。

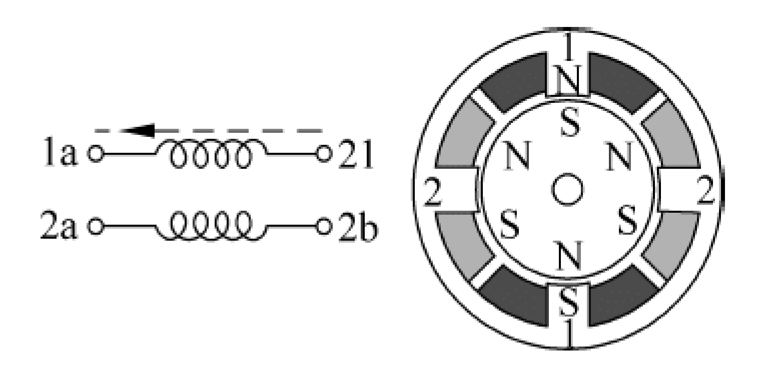


图 7.20　双极性两相步进电机结构

具体来说，双极性两相步进电机是指有两个线圈，四个接线端子。电流在两个线圈中可以正反向流动，所以叫做双极性。而单极性两相步进电机是指有两个线圈，但是有 5 条或 6 条线，即在每一个线圈的中间增加了一个抽头，即为 6 条线；把两个线圈的两根中间线并在一起，即为 5 条线。因为在一个线圈的中间有了抽头，电流就可以在一个线圈的一半走不同的流向，但这时只是用到电机线圈的一半而已。

双极性步进电机和单极性步进电机的产生是源于几十年前，因为步进电机驱动电路较贵，如果用双极性的驱动电路，比单极性的电路要贵很多，所以之前单极性的步进电机较多。随着步进电机驱动电路成本降低，双极性电机的应用多了起来。同时，双极性电机因为一个线圈都有正反向电流流过，应用效率高，单极性的则因为电机大部分时间只使用一半的线圈，所以效率较低。

2. 正反转控制

首先利用该例讨论步进电机的正反转控制。令图 7.19 中 CW/$\overline{\text{CCW}}$信号输入高电平，单击仿真运行键，可以看到步进电机沿顺时针方向以一定速度运转。然后停止仿真，将鼠标放在仿真图表上，单击鼠标右键，选择 Simulate Graph，如图 7.21，仿真图表更新后，得到波形如图 7.22。

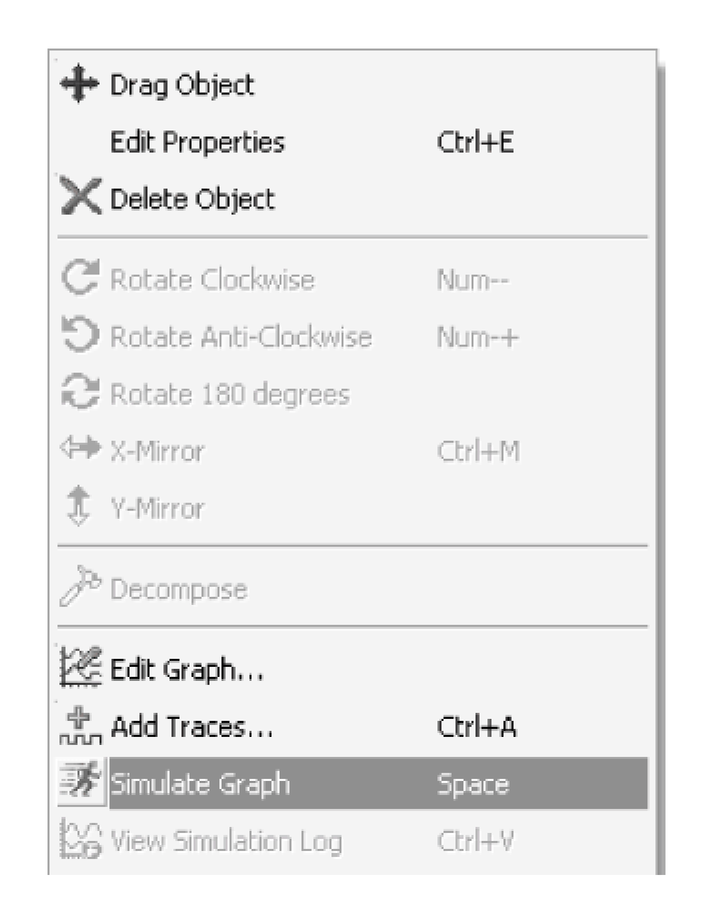


图 7.21　仿真图标属性莱单

为了波形清晰，这里只画出了 L297 输出给步进电机的四个信号及时钟信号。由图 7.22 波形可见，每个时钟周期，送出一拍控制信号，相序为：0101→1001→1010→0110→0101，属四相四拍控制方式。

令图 7.19 中 CW/$\overline{\text{CCW}}$信号输入低电平，单击仿真运行键，可以看到步进电机沿逆时针方向以一定速度运转。停止仿真，将鼠标放在仿真图表上，单击鼠标右键，再次选择 Simulate Graph，仿真图表更新后，得到图 7.23 波形。

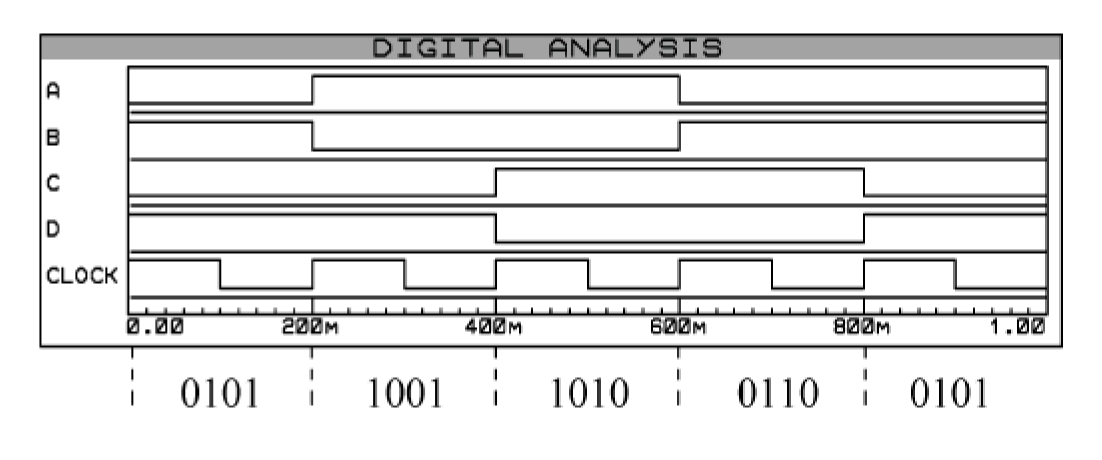


图 7.22　步进电机正转波形

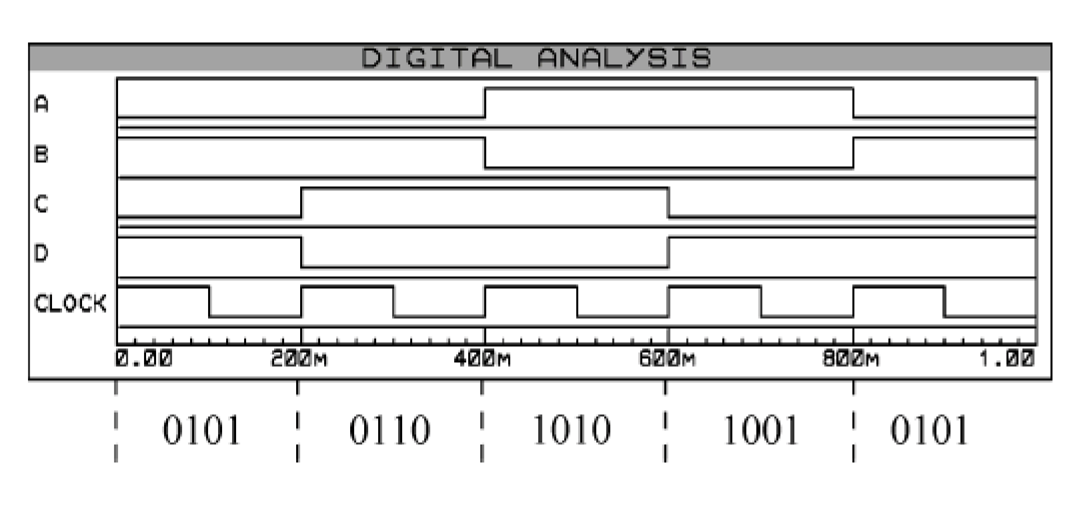


图 7.23 步进电机反转波形

由图 7.23 可见,电机逆时针旋转时,控制脉冲相序为:0101→0110→1010→1001→0101,与图 7.22 中电机顺时针旋转的控制脉冲相序相反。

因此,改变步进电机供电相序,即可改变其转动方向。

3. 不同工作方式控制

基于 L297/298 驱动器的双极性两相步进电机有三种不同的控制模式:半步模式、整步单相励磁模式、整步两相励磁模式。下面分别对半步模式和整步模式进行分析。

(1) 半步模式

双极性两相步进电机工作于半步模式时,其内部定子绕组的励磁是一相励磁与两相励磁交替进行的,此时步进电机每接受一个脉冲,只转过半个步距角。如果此种方式用于对四相步进电机的控制,即为四相八拍的控制方式,即单相供电与双相供电交替进行。当步进电机工作于半步模式时,步进电机所获得的转矩较常规值偏小。

将图 7.19 中 HALF/$\overline{\text{FULL}}$ 信号设置为高电平“1”,即半步模式,单击仿真运行键 ,可以看到步进电机沿顺时针方向运转,而且一拍转过 45°。接着,停止仿真,更新“仿真图表”,得到 L297 输出的 A、B、C、D 波形如图 7.24 所示。

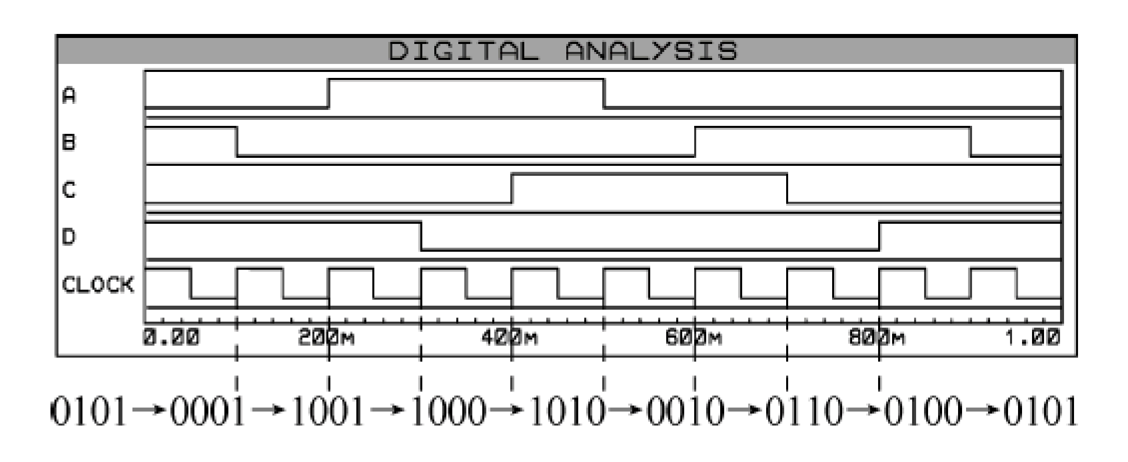


图 7.24 半步模式控制脉冲波形

由 7.24 图中波形可见,脉冲时序为:

0101→0001→1001→1000→1010→0010→0110→0100→0101

由以上时序可见,通电的相序是单相和双相间隔,而且每个时钟周期发一拍信号,每八拍构成一个周期,此时步距角为 45°。

(2) 整步模式

步进电机的整步模式分为整步单相励磁模式和整步两相励磁模式。整步模式是将上述半步模式八步程序中去掉间隔的状态而得到的。工作于整步单相励磁模式时,其内部定子绕组的励磁在任意时刻都是一相的,步进电机每接受一个脉冲,转过一个步距角。步进电机工作于整步单相励磁模式时,步进电机获得常规大小的转矩。

步进电机工作于整步两相励磁模式时，其内部定子绕组的励磁在任何一个时刻都是两相的，此时步进电机每接受一个脉冲，转过一个步距角。此时，步进电机获得的转矩相对前两种而言是最大的。

将图 7.19 中 HALF/$\overline{\text{FULL}}$信号设置为低电平模式，即整步模式，单击仿真运行键，可以看到步进电机沿顺时针方向运转，一拍转过 90°。接着，停止仿真，更新“仿真图表”，得到波形即为图 7.22。通电相序是：0101→1001→1010→0110→0101，一个周期四拍，步距角为 90°。每次都是两相同时通电，因此属于整步两相励磁模式。

可见，通过半步或整步工作模式的选择，可改变步距角的大小，步距角越小，通过步进电机进行的定位控制将越精确。

4. 控制程序设计

步进电动机控制程序的设计一般分为如下几个步骤：首先，根据系统总体设计，确定步进电机程序设计的具体要求(如是否充当脉冲分配器，怎样进行速度控制和方向控制等)；然后，设计程序的算法流程图；接着进行程序的设计、调试和测试；最后进行程序与系统硬件的联调。

在步进电动机的控制系统中，如果控制计算机不充当脉冲分配器的作用，则其主要进行速度、方向的控制，及其他协调控制等，由专门的硬件(脉冲分配器集成电路如：三洋公司的 PMM8713、PMM8723、PMM8714 等；包含脉冲分配器和电流斩波的控制器集成电路如：SGS 公司的 L297、L6506 等；将脉冲分配器、功率驱动、电流控制和保护电路都包括在内的驱动控制器集成电路如：东芝公司的 TB6560AHQ、Motorola 公司的 SAA1042(四相)和 Allegro 公司的 UCN5804(四相)等)来完成脉冲分配，如图 7.19 所示例子就是这样。

如果控制计算机需要完成脉冲的分配，则需要通过程序生成所需相序的脉冲信号，同时还需实现步进电机的走步数、转向及速度控制。即主要需解决如下几个问题。

(1) 步进电机的工作精度的确定

一般步进电机的精度为步进角的 3%～5%，且不累积。所以可以根据具体系统的要求选择适当步距角的步进电机，选择一定相序的工作模式，以达到控制精度的要求。

另外，为了提高步进电机的工作精度，可以采用细分技术。步进电机的细分技术实质上是一种电子阻尼技术，其主要目的是减弱或消除步进电机的低频振动，而提高电机的运转精度则只是细分技术的一个附带功能。比如对于步进角为 1.8°的两相混合式步进电机，如果细分驱动器的细分数设置为 4，那么电机的运转分辨率为每个脉冲 0.45°。电机的精度能否达到或接近 0.45°，还取决于细分驱动器的细分电流控制精度等其他因素。不同厂家的细分驱动器精度可能差别很大，细分数越大精度越难控制。

(2) 步进电机速度的调节

步进电机的速度控制，就是控制步进电机产生步进动作的时间，即控制步进电机各相绕组通电状态的切换时间，使步进电机按照给定的速度规律进行工作。输出脉冲通电相更换得越快，步进电机的转速越高。因此，如果要产生一个接近线性上升的加速过程，就可控制进给脉冲序列的时间间隔，由疏到密的命令步进电机产生步进动作。

(3) 硬件接口电路

由于步进电动机的驱动电流比较大，所以控制计算机与步进电机的连接都需要专门的接口电路及驱动电路。接口电路可以用缓冲器和锁存器组成，基本结构同第 3 章所述数字 I/O

接口部分。可以选用一些专用的 I/O 板卡，也可以选用并行 I/O 接口芯片，如 8255、8155 等。

如图 7.25 即为采用通用 I/O 板卡（PCI-1710HG 板卡）作为控制计算机与步进电机驱动控制接口的典型应用。PCI-1710HG 是一款基于 PCI 总线的多功能数据采集卡，它包含最常用的测量和控制功能，能够满足 12 位 A/D 转换、D/A 转换，可进行 16 路数字量输入、数字量输出，并具有计数器/定时器功能。具体用法在 3.5 节中有较详细介绍。

图 7.25 示意图中，利用控制计算机作为主控设备，PCI-1710HG 板卡通过 PCI 总线和主控机通信，控制对象为 x、y 双轴定位数控加工平台，利用接口板卡的数字量输出口 DO0～DO2 输出用于 x 轴步进电机控制的脉冲信号，此脉冲信号经驱动器，送往 x 轴的步进电机；数字量输出口 DO8～DO10 输出用于 y 轴步进电机控制的脉冲信号，此脉冲信号也经驱动器，送往 y 轴的步进电机。通过上位计算机的插补算法、脉冲分配、速度控制，协调 x、y 轴步进电机，实现精确的加工与定位。

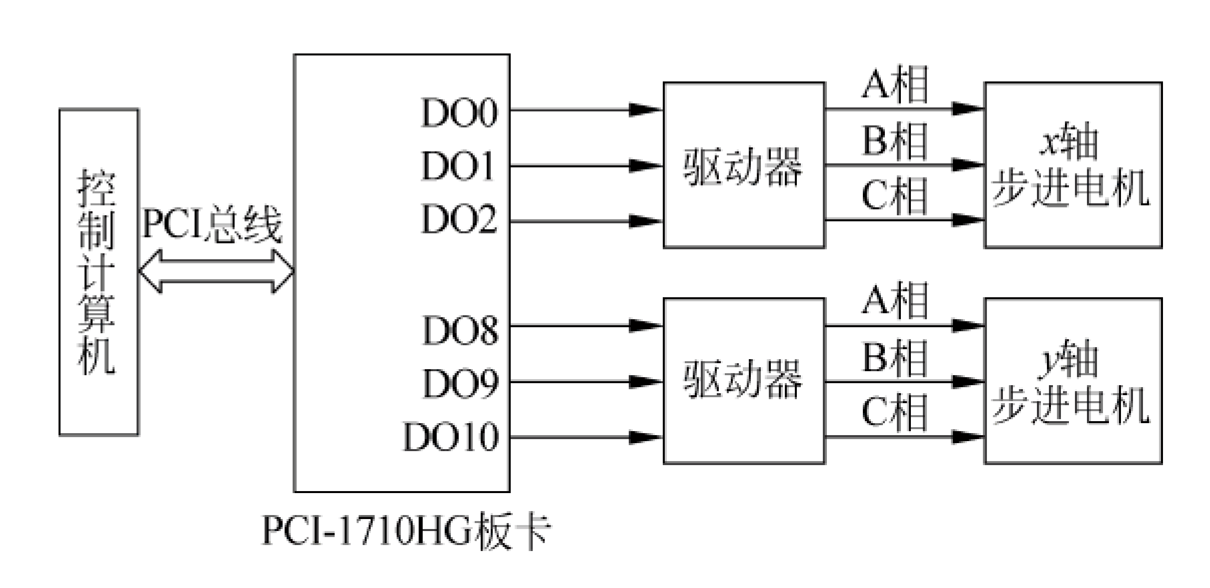


图 7.25 步进电机控制系统硬件接口示意图

(4) 实现插补算法的走步控制

下面以图 7.25 所示的基于三相步进电机的 x、y 双轴定位数控加工平台控制系统为例，讨论步进电机的走步控制。在数控加工过程中，需首先根据 7.2 节插补算法进行插补计算，然后实时控制步进电机向 $+x$、$-x$、$+y$、$-y$ 方向进行走步，每进给一步，工作台移动一个脉冲当量。

这里，控制计算机用查表的方式来完成脉冲分配，设定步进电机以三相单双六拍模式进行工作，因此，首先建立三相步进电机控制的输出字表，如表 7.5 所示。

表 7.5 三相步进电机控制的输出字表

x 轴步进电机输出字表		y 轴步进电机输出字表	
存储地址	DO7～DO0	存储地址	DO15～DO8
$DX1$	00000001＝01H	$DY1$	00000001＝01H
$DX2$	00000011＝03H	$DY2$	00000011＝03H
$DX3$	00000010＝02H	$DY3$	00000010＝02H
$DX4$	00000110＝06H	$DY4$	00000110＝06H
$DX5$	00000100＝04H	$DY5$	00000100＝04H
$DX6$	00000101＝05H	$DY6$	00000101＝05H

用 DX 和 DY 分别表示 x 轴和 y 轴步进电机输出字表的取数地址指针；用 $ZF=1$、2、3、4 分别表示 $+x$、$-x$、$+y$、$-y$ 走步方向。结合图 7.9 的直线插补程序流程图或图 7.14 的圆弧插补程序流程图，做进给程序的流程图如图 7.26 所示。

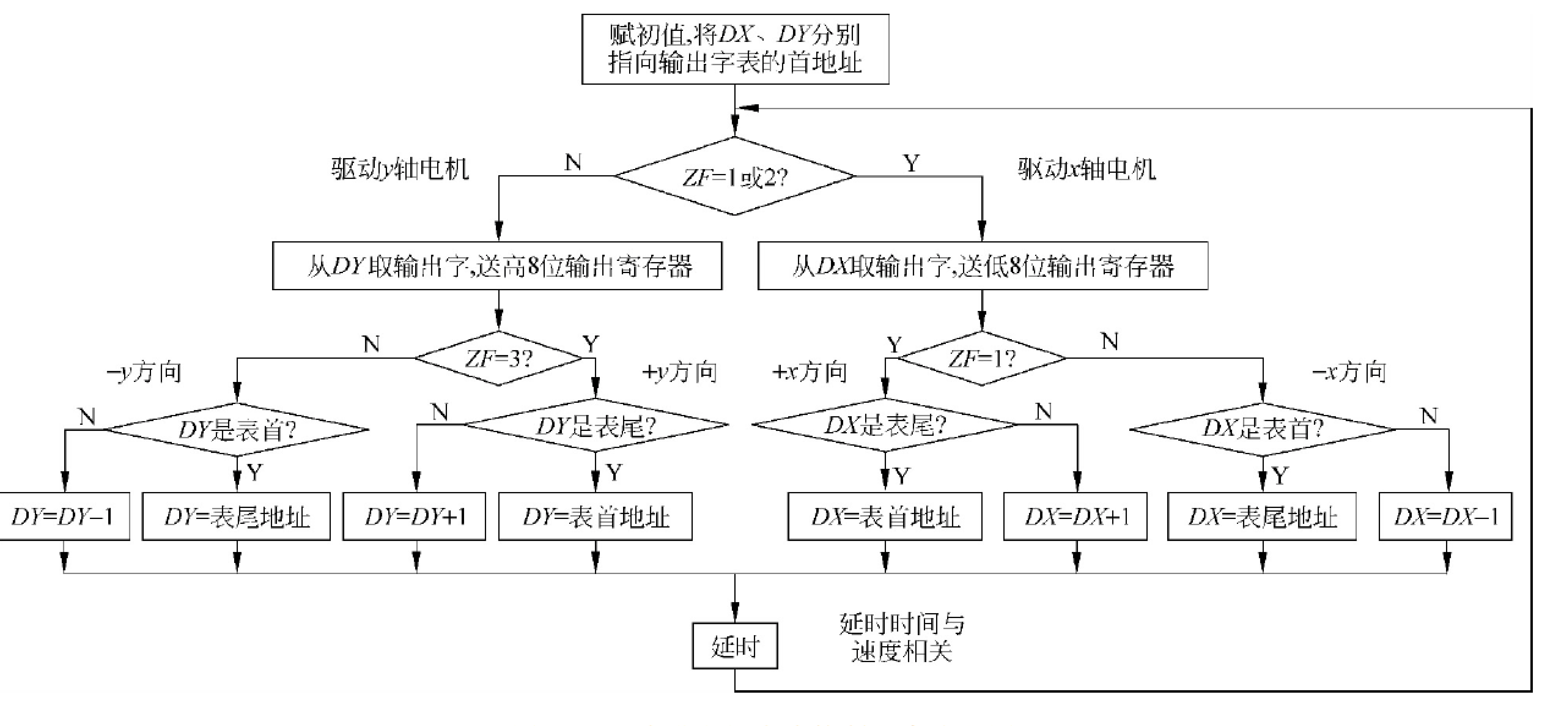


图 7.26　步进电机走步控制程序流程图

其中延时时间长短将决定控制脉冲的频率，从而决定步进电机的运行速度。另外，控制计算机进行脉冲分配的方法很多，例如可以采用定时器产生所需脉冲等。

7.3.3 步进电机的振动及失步

步进电机低速转动时会产生振动，噪声较大；电机运转时运转的步数如果不等于理论上的步数，就会造成失步，造成走步不准确。下面对这些问题一一讨论。

1. 步进电机的振动

步进电机的振动是其固有的缺点，尤其在转速比较低的时候，一般可采用以下方案来克服。

(1) 如步进电机正好工作在共振区，可通过改变减速比等机械传动方式避开共振区。

(2) 采用带有细分功能的驱动器。

(3) 换成步距角更小的步进电机，如三相或五相步进电机。

(4) 换成交流伺服电机。

(5) 在电机轴上加磁性阻尼器。

其中，细分的原理与方法在7.3.3节中已说明。

2. 步进电机的失步

步进电机的失步往往是因为步进电机中产生的同步力矩无法使转子速度跟随定子磁场的旋转速度所造成的。失步产生的原因有很多，主要有以下几个方面。

(1) 提供给步进电机的转矩不够，尤其当驱动脉冲频率达到或超过某临界值时。因为步进电机的动态输出转矩随着驱动信号的频率的上升而下降。可通过在额定电流范围之内适当增加驱动电流，或适当提高驱动电压来提高电磁转矩；也可以改用转矩大的步进电机，使步进电机需要克服的转矩减小；或者适当降低电机的运行频率，以便提高电机的输出转矩。

(2) 步进电机起动时发生失步。由于一般情况下系统的极限启动频率是比较低的，而要求的运行速度往往较高，如果系统以运行的速度直接起动，因为该频率已超过极限启动频率而不能正常起动，就会发生丢步或不能起动。应设置合理的加速时间，使电机从低速平稳上升到某个速度。

(3) 步进电机发生共振也是引起失步的一个原因。如果步进电机在连续运行状态时，控制脉冲的频率等于步进电机的固有频率，就将发生共振。可通过改变减速比等机械传动方式避开共振区来解决，或减小步进电机的驱动电流，也可采用细分驱动方法和阻尼方法解决。

3. 步进电机的细分驱动

上述提到在需要改善电机性能，如振动、噪声、失步等，都可以采取细分的方法来解决；另外，在步进电机步距角不能满足使用的条件下，也可采用细分驱动器来驱动步进电机。细分驱动器的原理是通过改变通电相相电流的大小，以改变合成磁场的夹角来控制步进电机

的运转。

如果按照环形分配器决定的分配方式驱动步进电机，控制电机各相绕组的导通或截止，从而产生使电机步进所需的旋转磁势，拖动转子步进旋转，则对于一个确定的步进电机来说，步距角只有两种，即整步方式或半步方式，而步距角由电机结构所确定。如果要求步进电机有更小的步距角，更高的分辨率，或者为了解决电机振动、噪声等情况，可以当每次进行脉冲切换时，在一拍中，通电相的电流不是一次达到最大值，而是分为多次，每次只使绕组电流增加一部分，则电机的合成磁势也只旋转步距角的一部分，因此，转子的每步运行也只有步距角的一部分。同理，绕组电流的下降也分多次完成。这样步进电动机原来的一个步距，便被分成许多微步运动来完成，即实现了步距的细分。

以两相步进电机为例，由于步进电机的运行依赖于电机内部定子绕组产生的合磁场，而每个绕组所产生的磁场的大小和方向又依赖于该绕组中电流的大小和方向。因此可以通过合理地选择两个绕组中的电流，实现步进电机的恒定转矩的细分驱动控制。如图 7.27 所示是两相步进电机细分驱动的原理。其中，Φ_A 表示 A 相通电时所产生的磁场，Φ_B 表示 B 相通电时所产生的磁场。如果将 A 相通电时磁场方向定义为起始位置，此时磁场唯一由 A 相电流产生，即 $\Phi_0=\Phi_{A0}$；接着逐渐减小 A 相电流，增大 B 相通电电流，依次可得到磁场 Φ_1、Φ_2、Φ_3，直到只有 B 相通电，得到 $\Phi_4=\Phi_{B0}$；而 Φ_1、Φ_2、Φ_3 是由 A、B 两相产生的合成磁场。从只有 A 相正向通电变为只有 B 相正向通电，磁场沿顺时针方向转了 90°，这是步进电机整步运行的情况；如果 A、B 相同时通电，且电流相同，合成磁场为图 7.27 中 Φ_2 所示方向，这是步进电机半步运行的情况。Φ_1、Φ_3 即为细分的结果，即在半步方式的每两步之间插入一步。

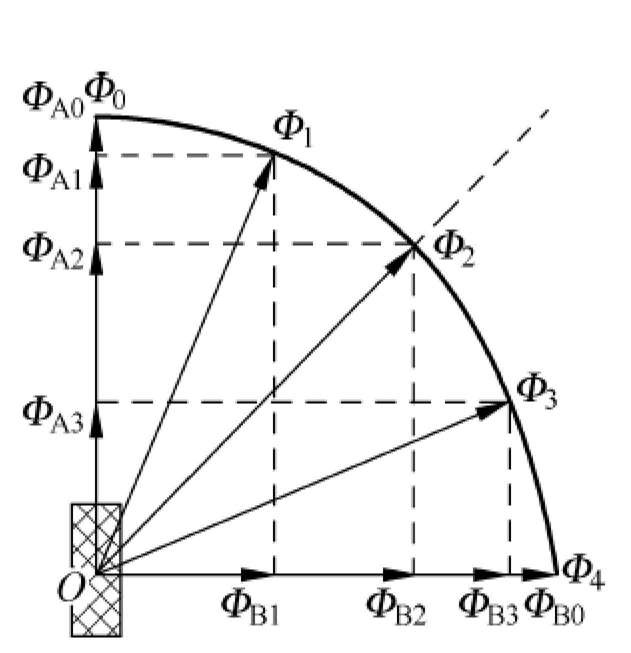


图 7.27　两相步进电机细分驱动原理

目前有很多步进电机驱动专用芯片都具有细分的功能，例如，TA8435 是东芝公司生产的单片正弦细分二相步进电机驱动专用芯片，下面简要介绍一下该芯片的特点和细分应用电路设计。

TA8435 芯片的特点有以下几个方面。

(1) 工作电压范围宽(10～40V)。

(2) 输出电流可达 1.5A(平均)和 2.5A(峰值)。

(3) 具有整步、半步、1/4 细分、1/8 细分 4 种运行方式可供选择。

(4) 采用脉宽调制式斩波驱动方式。

(5) 具有正/反转控制功能。

(6) 带有复位和使能引脚。

(7) 可选择使用单时钟输入或双时钟输入。

TA8435 主要由 1 个解码器、2 个桥式驱动电路、2 个输出电流控制电路、2 个最大电流限制电路、1 个斩波器等功能模块组成。共有 25 个引脚，引脚功能见表 7.6。

表 7.6 TA8435 引脚功能

引脚号	引 脚 名 称	说 明
1	S-GND	信号地
2	$\overline{\text{RESET}}$	复位
3	$\overline{\text{ENABLE}}$	使能端，高电平禁止工作
4	OSC	该引脚外接电容决定芯片内部驱动级的斩波频率(15～80kHz)，计算公式为：$f_{osc}=1/5.15\times C_{osc}$，式中，$C_{osc}$的单位为$\mu$F，$f_{osc}$的单位为 kHz
5	CW/$\overline{\text{CCW}}$	正反转控制引脚
6、7	CK2、CK1	时钟输入端，可选择单时钟输入或双时钟输入，最大时钟输入频率为 5kHz
8、9	M1、M2	工作方式设定引脚，00 表示步进电机工作在整步方式，10 为半步方式，01 为 1/4 细分方式，11 为 1/8 细分方式
10	REFIN	VNF 输入控制，接高电平时 VNF 为 0.8V，接低电平时 VNF 为 0.5V；即 REFIN＝1 时，NFA、NFB 输出电压为 0.8V；REFIN＝0 时，NFA、NFB 输出电压为 0.5V
11	$\overline{\text{MO}}$	输出监视，用于监视输出电流峰值
12	NC	没有连接
13	VCC	逻辑电路供电引脚，一般为 5V
14	NC	没有连接
15、24	VMB、VMA	B 相和 A 相负载电源端
16、19	$\overline{\text{B}}$、B	B 相输出引脚
17、22	PG-B、PG-A	B 相和 A 相负载地
18、21	NFB、NFA	B 相和 A 相电流检测端，由该引脚外接电阻 RNF 和 REFIN 引脚控制的输出电流为：I_O＝VNF/RNF
20、23	$\overline{\text{A}}$、A	A 相输出引脚
25	NC	没有连接

图 7.28 是单片机与 TA8435 相连控制步进电机的典型应用原理图。引脚 M1 和 M2 决定电机的转动方式；CW/$\overline{\text{CCW}}$控制电机转动方向，CK1、CK2 时钟输入的最大频率不能超过 5kHz，控制时钟的频率，即可控制电机转动速率；REFIN 为高电平时，NFA 和 NFB 的输出电压为 0.8V，REFIN 为低电平时，NFA 和 NFB 输出电压为 0.5V，这两个引脚控制步进电机输入电流，电流大小与 NF 端外接电阻关系式为：I_O＝VNF/RNF；二极管 D1～D4 用来进行绕组泄流。步进电机按二相双极性使用，四相按二相使用时可以提高步进电机的输出转矩。

细分主要是用于改善电机的运行性能，使运动更加平稳，振动减小，噪声降低，且不易失步，同时使步进电动机的分辨力得以提高。由于细分驱动器要精确控制电机的相电流，所以对驱动器要有相当高的技术要求和工艺要求。

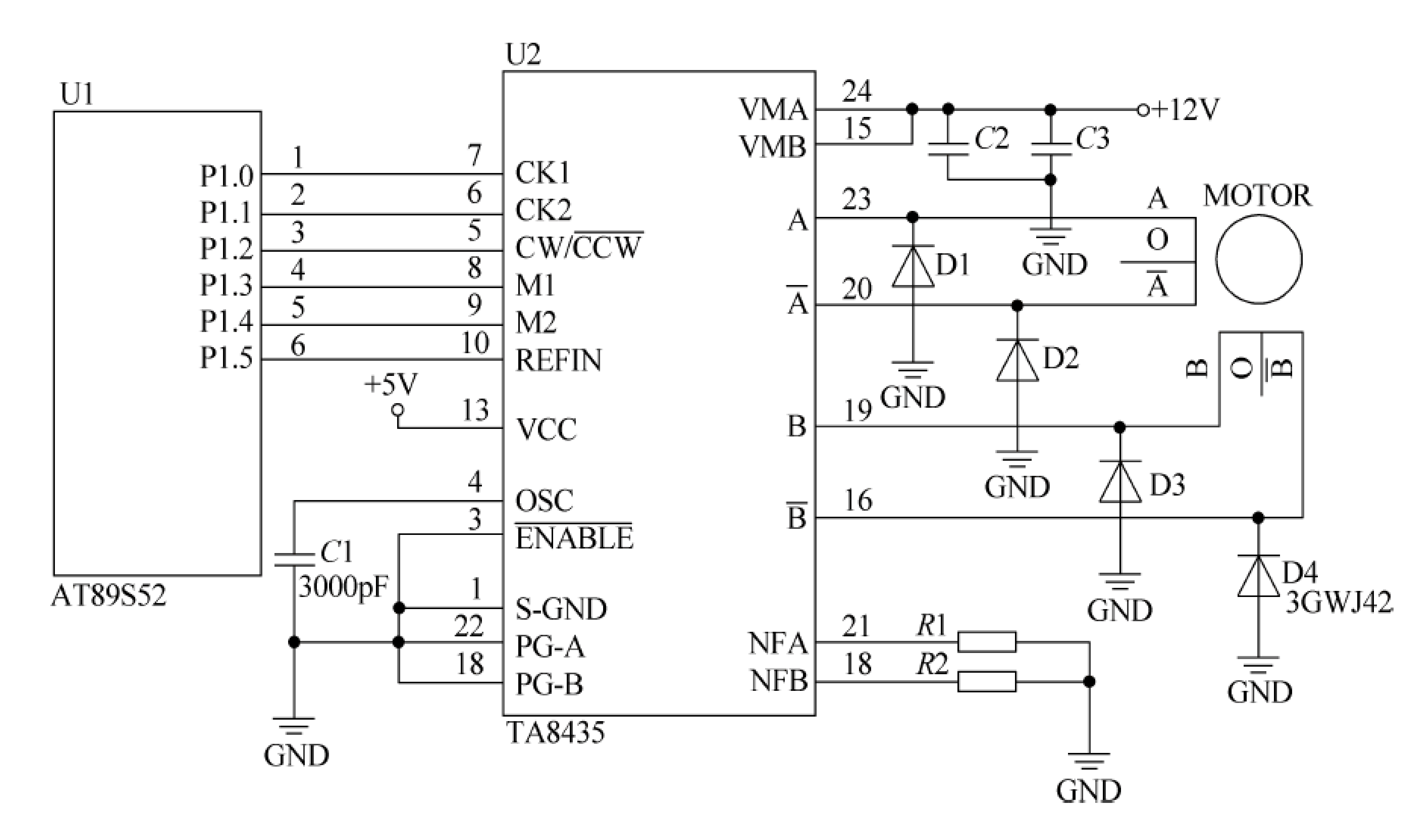


图 7.28　TA8435 典型应用原理图

7.4　其他数字程序控制驱动设备

上述着重讲了以步进电机为驱动设备的数字程序控制技术，以及系统实现的方法。更详细的内容可以在伺服控制相关课程中了解。除此之外，常用的数字程序驱动设备还有很多，如表 7.7 所示。如果按照在数控系统中的应用分类，进给系统因为控制的重点在于精确定位，常使用步进电机、直流伺服电机、交流伺服电机等驱动器，而与之相配套的驱动电源包括步进电机驱动器、直流伺服驱动器和交流伺服驱动器；主轴驱动系统的控制重点在于速度控制，因此常选用直流和交流主轴电动机，其驱动电源主要是由晶闸管构成的驱动电路，或是采用变频器实现控制。

表 7.7　常用的数字程序驱动设备

用　　途	驱 动 电 机	驱 动 电 源
进给控制	步进电动机	低压供电、高低压切换、恒流斩波等(步进驱动器)
	直流伺服电动机	脉宽调制、晶闸管控制等(直流伺服驱动器)
	交流伺服电动机	变频控制(伺服控制器)
主轴驱动	直流主轴电动机	晶闸管控制
	交流主轴电动机	变频控制(变频器)

习题

1. 什么是数字程序控制？数字程序控制可分为哪些种类？
2. 什么是逐点比较法插补？直线插补计算过程和圆弧插补计算过程各有哪几个步骤？
3. 设加工第一象限直线 OA，起点 $O(0,0)$，终点 $A(5,3)$。要求：

(1) 按逐点比较法直线插补进行列表计算;

(2) 做出走步轨迹图,并标明进给方向和步数;

(3) 试作出算法程序流程图。

4. 设加工第一象限一条逆圆弧 AB,起点为点 $A(5,0)$,终点为 $B(4,3)$。要求:

(1) 试按逐点比较法列表进行插补计算;

(2) 做出走步的轨迹图,并标明进给方向和步数;

(3) 给出算法程序流程图。

5. 一台四相步进电动机,采用四相八拍脉冲分配方式,设转子有 50 个齿,脉冲频率为 $f=600\text{Hz}$,丝杠导程为 $t=10\text{mm}$,减速装置的传动比 $i=10$。求:

(1) 写出一个循环的通电顺序;

(2) 求步进电动机步距角;

(3) 求步进电动机转速;

(4) 求步进电机的脉冲当量。

第8章 计算机控制系统软件设计

计算机控制系统是由硬件和软件两大部分组成的，只有硬件的计算机称为裸机，它只为计算机控制系统提供了物质基础，不能实现任何功能，因此必须为其提供或研制软件，才能实现对生产过程的控制。软件是计算机控制系统的核心，是计算机控制系统的程序算法系统，只有通过软件和硬件的有机结合，计算机才能实现各种控制策略、算法，才能充分发挥计算机的优势，使计算机控制系统具有更高的性价比，达到控制目标，完成控制任务。本章主要介绍计算机控制系统软件的设计方法，重点介绍软件的组成和功能，软件系统中的数据处理方法。

8.1 计算机控制系统软件体系结构

8.1.1 软件分类

软件在计算机系统中和硬件是相互依存的，它是程序、数据和相关文档的集合。计算机控制系统软件分为系统软件和应用软件两种。

1. 系统软件

系统软件用来管理计算机系统的所有资源，并以尽可能简便的形式向用户提供使用资源的服务，主要功能是调度、监控和维护计算机系统，负责管理计算机系统中各种独立的硬件，使得它们可以协调工作。系统软件使得计算机使用者和其他软件将计算机当作一个整体而不需要顾及到底层每个硬件是如何工作的。

一般来讲，系统软件包括操作系统和一系列基本的工具（比如编译器、数据库管理、存储器格式化、文件系统管理、驱动管理、网络连接等方面的工具），是支持计算机系统正常运行并实现用户操作的那部分软件。比如常用的计算机操作系统有 Windows、Linux、DOS、UNIX 等，语言处理程序有汇编语言汇编器、C 语言编译器及链接器等。

2. 应用软件

应用软件是设计人员针对某个生产过程或者解决某类应用问题而专门研制的控制和管理程序，主要包括文字处理软件、图形图像处理软件、系统开发软件、控制类软件等。在计算机控制系统应用软件中主要包括过程输入输出程序、过程控制程序、人机接口程序、数据处

理程序、报警程序、打印显示程序等。

8.1.2 计算机控制系统软件要求及功能

在整个计算机控制系统中，所需要的硬件大部分都可以直接从市场购买到，因此研制一个计算机控制系统，软件研制的工作量往往要大于硬件，所以计算机控制系统设计，很大程度上可以说是软件设计，软件成为影响整个控制系统性能的关键。

1. 控制系统对软件的要求

(1) 实时性。由于工业控制是实时控制系统，要求被控对象在允许的时间间隔内完成对系统的控制、计算和处理等任务，特别在控制任务比较复杂时，程序的执行时间是控制系统应用软件设计必须顾及的一个问题。应特别注意采样周期、控制周期与中断周期在实时性方面能否满足要求。因此除了在硬件上采取必要的措施之外，还必须在软件设计上加以考虑，主要是提高软件的响应和处理速度。比如采用汇编语言提高程序执行速度，在满足系统控制要求的前提下降低采样频率减轻系统的处理负担。

(2) 多任务和多线程。现代工业生产中的综合自动化系统不仅包括各种简单和复杂的控制系统，还包括生产管理、调度和财务管理等系统，因此应用软件所面临的对象不再是单一任务或线程，而是比较复杂的多任务系统。因此，多任务实时软件系统是控制和管理复杂对象的有效手段，也是目前控制软件研究的主要方向。

(3) 灵活性和通用性。在计算机控制系统应用程序设计中，为了提高设计效率，节省内存，提高程序的复用性，使软件具有较强的适应能力，要求其具有一定的灵活性和通用性。在软件设计上可以采用模块化和结构化设计，尽量将共同的程序编成子程序等设计措施。

(4) 可靠性和抗干扰性。工业控制现场环境一般比较复杂恶劣，干扰源比较多，并且控制系统的高可靠性又是必需的。因此，在计算机控制系统设计过程中，除了考虑硬件的可靠性和抗干扰性外，还必须顾及软件的可靠性和抗干扰性，两者的可靠性同等重要。构成双通道的抗干扰防线，无疑将大大提高计算机控制系统的可靠性。在软件设计中可以设置检测与诊断程序，采用冗余技术和软件陷阱等技术来提高其可靠性和抗干扰性。

2. 计算机控制系统软件功能

计算机控制系统输入输出、运算控制和显示操作等功能的实现，除了有相应的硬件和算法之外，还要有相应的软件支撑，硬件、算法和软件的有机结合才能构成完整的计算机控制系统。其应用软件一般都具有以下功能。

(1) 计算机控制系统首要工作是完成实时的数据采集和实时控制输出。其在输入软件的控制下采集现场控制设备的状态及过程参数，在输出软件的支持下进行闭环控制输出，以达到完成控制任务的目的。

(2) 计算机控制系统软件的核心是实时控制决策和策略。其对输入软件中采集到的被控量进行分析和处理，并按已定的控制规律，决定将要采取的控制行为，为控制系统提供可供选择的控制策略方案。

(3) 计算机控制系统软件的其他附属功能也是一个完整的控制系统必不可少的。其主要有：使来自现场设备的数据与计算机图形监控画面上的各元素关联起来的画面显示功

能；各类数据报表的生成和打印输出的功能；存储历史数据并支持历史数据查询的数据存储功能；处理数据报警以及系统报警的报警监视功能；各控制操作站以及子系统间的数据传输通信功能；系统的自诊断、掉电处理以及其他附属功能。

由于计算机控制系统的特点，对其控制软件提出的功能和指标要求比一般的软件要求要高出很多，因此对计算机控制系统软件的设计也相应提出了较高的要求。

8.1.3　应用软件的设计流程及方法

计算机控制系统的软件设计主要是选择系统软件和应用软件设计。前者主要是选择合适的操作系统及其配套的软件，后者主要是根据具体的控制系统任务要求开发设计应用软件。系统软件由开发商提供不需要用户自行设计，用户只需要知道它的应用，将其作为一种编程及开发的工具。而应用软件主要是根据不同的应用场合由用户自行编制，对于不同的控制任务和生产对象，其一般也不同，因此通用性差，但大致都是按照以下几个步骤来设计。

软件设计的过程如图 8.1 所示。首先是根据用户的要求对系统进行分析，即依据系统的具体任务指标，确定控制方案，定义输入和输出，选择控制算法等。随后就是编写系统程序规范的说明文档，然后就是软件设计阶段，主要是把说明文档中所定义的问题用程序的方式对其进行描述，比如常用的流程图等。软件设计之后就需要把设计框图变成计算机能处理接收的指令，这就是编码与实现阶段，主要是用高级语言或者汇编语言来编写实现。最后进行程序的测试、查错和调试阶段，通过测试查错来发现程序中的错误，修改相应的错误，直到程序正确实现所设计的功能，完成预定的任务。

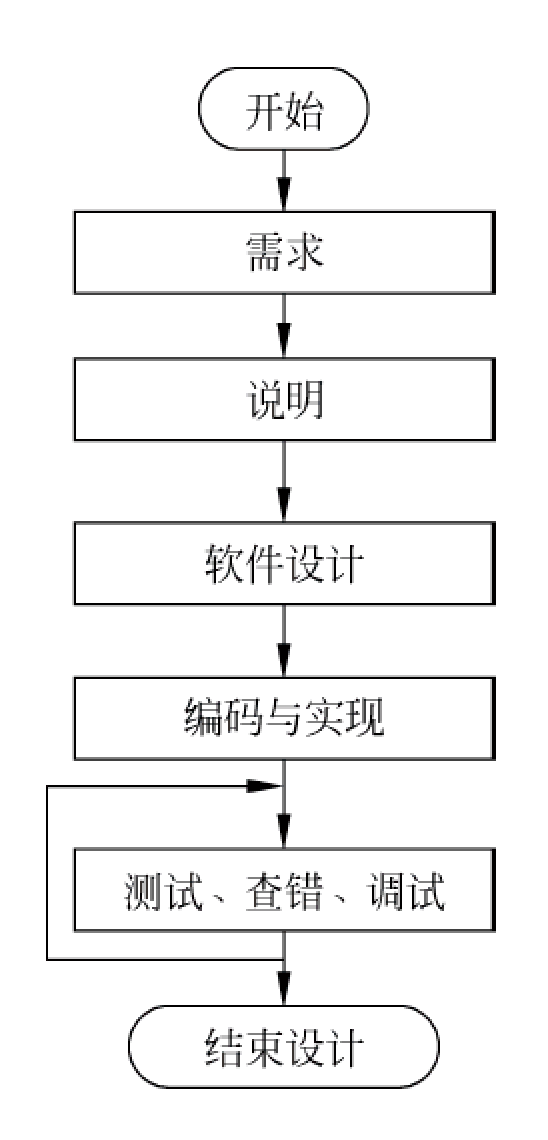


图 8.1　应用软件设计流程图

计算机控制系统软件设计的原则是用比较短的时间和比较少的资源开销设计出功能完备、阅读容易、修改方便的程序，因此必须采用科学的软件设计方法，常用的程序设计方法有模块化设计和结构化设计等方法。

(1) 模块化程序设计。模块化设计的基本思想是把一个复杂的系统分解为若干个功能模块，每个功能模块相对独立，执行单一的功能，并且具有单入口单出口结构，结构清晰，接口简单，每个模块分别进行独立的编程设计之后再装配在一起，从而连接完成整个系统软件的设计。这样的设计方法也使系统易于维护和进行功能的扩充。

顶层模块调用分解好的它的下层模块以实现完整功能，每个下层模块再调用更下层的模块，底层模块完成最具体的功能任务。模块化程序设计的方法有自底向上模块化设计和自顶向下模块化设计。

自底向上模块化设计就是首先对分解任务后处于最底层的模块进行编码、测试和调试。在这些模块正常工作后，就可以用它们来开发较高层次的模块。这种方法是汇编语言常用的方法。

自顶向下模块化设计就是首先对最高层的模块进行编码、测试和调试。为了测试这些最高层的模块，可以用“节点”来代替还没有编码的较低层模块，这些“节点”的输入和输出要

满足程序的说明部分要求,但是功能少得多。这种方法一般适合于用高级语言来设计程序。

上述两种程序设计方法各有优缺点。在自底向上模块化设计中,高层模块设计中的根本错误可能要很晚才能发现。在自顶向下模块化设计中,整体程序的大小和性能往往要开发关键性的底层模块时才会表现出来。所以在实际设计中,最好将两种方法结合起来使用,先开发高层模块和关键性的底层模块,并用“节点”来代替以后开发的不太重要的模块。

(2) 结构化程序设计。软件的结构化设计主要是随着系统规模和复杂度的不断增大而提出的,结构化程序设计采用自顶向下逐步求精的设计方法和单入口单出口的控制结构。自顶向下逐步求精的设计方法符合抽象和分解的原则,是解决比较复杂问题常用的方法。在设计一个模块实现算法时先整体后局部,先抽象后具体,再通过逐步细化,最后得到详细的实现算法。单入口单出口的控制结构可以使程序有着良好的结构,从而增加了程序的可读性。

对于单入口单出口的程序只用三种最基本的控制结构就能实现,这三种基本的控制结构是“顺序”、“选择”和“循环”,其基本结构流程图如图 8.2 所示,其中循环结构又分为先判断再循环的当型循环和先循环后判断的直到型循环两种形式。

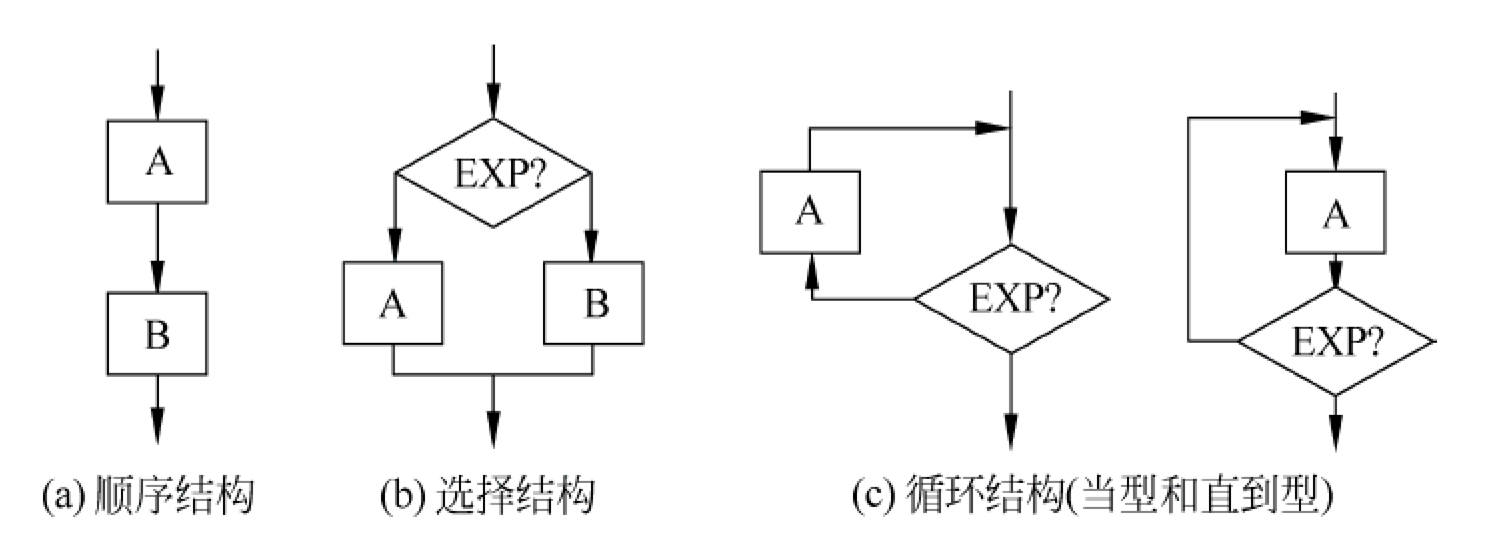


图 8.2 程序基本控制结构流程图

8.2 系统应用程序设计

在计算机控制系统中,除了操作系统之外,还有大量的应用程序。所谓应用程序就是针对控制系统的具体要求,由用户自己设计的程序。计算机控制系统的输入输出功能、运算控制功能以及人机接口显示操作等功能的实现,除了相应的硬件系统和算法之外,还必须要有相应的软件做支撑,通过硬件、算法和软件的有机结合才能构成完整的计算机控制系统。由于各种生产对象种类繁多,对控制的具体要求也不同,所以计算机控制系统的应用程序也是多种多样的。本节主要介绍计算机控制系统中常用的应用程序设计方法。

8.2.1 输入输出软件

计算机控制系统的输入和输出单元主要由各种类型的板卡或模块组成,它们是控制系统主机和生产过程之间设置的信号传递及变换的连接通道。它主要有两个作用,其一是将生产过程的信号通过采样处理变成主机能够接受和识别的代码;其二是将主机处理后的控制命令或者数据变换成执行机构能够识别的控制信号。与输入输出单元相配套的输入输出软件主要有输入输出接口程序和输入输出驱动程序以及对应的实时数据库。其软件结构如图 8.3 所示。

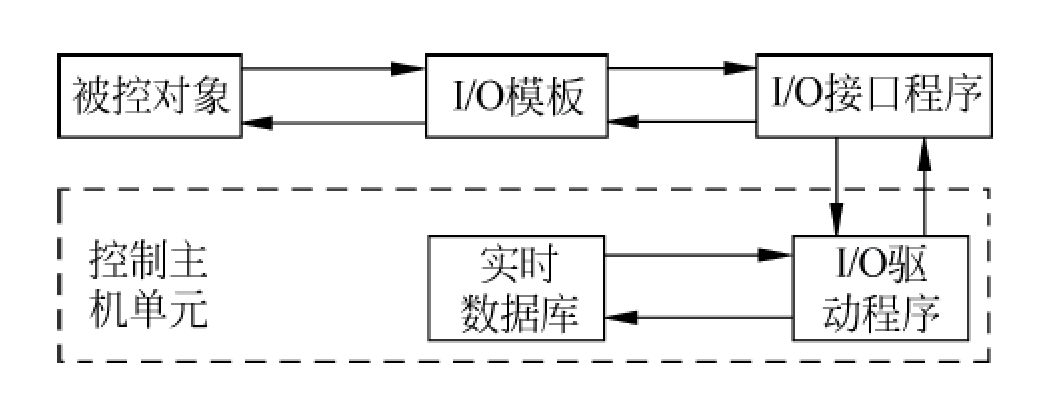


图 8.3 输入输出软件结构图

1. 输入输出接口程序

输入输出接口(I/O)程序是针对具体的I/O板卡或者模块而开发编写的程序。常用的I/O模板包括模拟量输入/输出(AI/AO)模板、数字量输入/输出(DI/DO)模板、脉冲量输入/输出模板以及混合功能的接口模板,每类模板按照信号的类型以及功能的不同又可以分为不同的信号模块。不同信号类型的I/O模板所对应的I/O接口程序也不一样,接口程序一般是用汇编语言编写的最初级的程序,位于I/O单元内。并且目前的绝大部分I/O接口都是采用可编程的接口芯片,它们的工作方式可以通过编程来设置。

2. 输入输出驱动程序

I/O驱动程序是针对模板和控制主机之间的数据交换或通信而编写的程序,是I/O模板和控制主机通信的桥梁,位于控制主机内。其主要功能是可以接收来自I/O单元的原始数据,同时还可以向I/O单元发送相应的控制命令或参数,并且可以与控制主机上的实时数据库进行无缝连接。

I/O单元和控制主机的通信方式常用的有板卡方式、串行通信方式、OPC(OLE for Process Control,用于过程控制的OLE)方式以及网络通信方式等。例如北京亚控公司的组态王支持的通信方式就有串口通信、数据采集板、DDE通信、人机界面卡、网络模块以及OPC。不同I/O单元以及不同通信方式的驱动程序一般也不一样,驱动程序不需要用户编写,一般的模板都自带有相应的驱动程序。

3. 实时数据库

实时数据库是计算机监视和控制系统的核心之一,一般都位于控制主机内,主要是建立存放数据点、统计处理、累计处理以及历史数据存储等功能。I/O驱动程序对来自I/O单元的原始数据进行有效性检查以及相应的处理后,就送到实时数据库建立数据点,只有变成实时数据库中的有效数据,才能被其他的程序和软件来方便的访问和使用。同样,实时数据库中不仅有接收来自I/O单元的数据,还有经过控制运算之后要发送到I/O单元的数据,比如操作参数或控制命令等。

实时数据库设计主要包括实时数据库结构设计和实时监控管理程序设计两部分。实时数据库结构设计主要根据计算机监视和控制系统的特点和要求进行,监控管理程序负责实时数据库的产生,处理其他程序任务对实时数据库的实时请求以及报警等对外界操作的响应。

输入输出软件是计算机和现场传送相应信息和命令的桥梁,所以必须根据控制要求合理的编写或者配置。比如针对使用研华公司的数据采集卡PCL-818L来采集数据,除了硬

件的正确配置和安装外,还要到研华公司官方网站 www.advantech.com.cn 找到并下载 PCL818.exe 驱动程序和设备管理程序 DevMgr.exe 等,然后根据安装向导完成驱动程序和配置管理软件的安装,只有正确的安装相应的程序之后,才能在控制主机中识别和使用,当用户的计算机上已经安装好某个产品的驱动程序后,设备管理软件支持的设备列表前将没有红色叉号,说明该设备驱动程序已经安装成功。配置好板卡 PCL-818L 驱动程序的研华公司的设备管理器界面如图 8.4 所示。在图 8.4 中,还可以利用板卡附带的接口测试程序对板卡的各项功能进行测试。通过驱动程序和设备接口函数,其数据和状态还可以被其他的程序和软件(比如 VC、VB 等)使用。



图 8.4 设备管理器界面

8.2.2 控制策略软件

控制策略是为控制系统提供实现控制目标任务的控制算法,工业生产过程中计算机控制系统控制效果的优劣,很重要的问题之一是由控制算法决定的。算法是建立在控制对象的数学模型基础上,即描述各控制输入量和输出量之间的数学关系。因此,首先应正确建立系统的数学模型,并根据数学模型确定系统的控制算法,然后由软件设计者在一定的硬件和软件环境平台上开发控制算法软件,常用的工业控制计算机的系统软件、算法语言及其配套的开发软件市场也非常成熟,种类也比较多,为设计者提供了灵活的开发工具。比如可以使用 C、VC、C++等来编写实现控制算法。

由于控制系统的种类繁多,所以采用的控制算法也各不相同,每个控制系统都有特定的控制规律,并且也都有相应的控制算法。比如在直接数字控制系统中,常采用的是 PID 控制算法及其一系列改进型算法;在数字机床控制系统中,常采用的是逐点比较法控制算法;在位置随动系统中主要采用实现最少拍控制的控制算法。另外还有模糊控制、批量控制、智能控制、最优控制等控制算法。这些控制算法在相关的文献中都有详细的论述。

控制算法直接影响控制系统的调节品质,是决定整个控制系统性能指标优劣的关键。在进行系统设计时,究竟选择哪一种控制算法,主要取决于控制系统的特性和要求达到的控制速度、控制精度和系统稳定性的性能指标。

【例 8.1】 计算机控制是一种采样控制,它只能根据采样时刻的偏差值来计算控制量。因此连续 PID 控制算法不能直接使用,需要采用离散化的方法使用数字 PID 控制器。比如对于控制对象为:$G(s)=\frac{523\,500}{s^3+87.35s^2+10\,470s}$的被控对象,采样时间为 1ms,采用双线性 Z 变换进行离散化,试针对离散系统的阶跃输入信号,设计离散 PID 控制器,并观察输入和输出之间的跟踪效果。

可以用下面的 MATLAB 语句来实现:

```
%PID Controller
clear all;
close all;
ts = 0.001;
sys = tf(5.235e005,[1,87.35,1.047e004,0]);
dsys = c2d(sys,ts,'z');
[num,den] = tfdata(dsys,'v');
u_1 = 0.0;u_2 = 0.0;u_3 = 0.0;
y_1 = 0.0;y_2 = 0.0;y_3 = 0.0;
x = [0,0,0]';
error_1 = 0;
for k = 1:1:1500
time(k) = k * ts;
    kp = 0.50;ki = 0.001;kd = 0.001;
    rin(k) = 1;                                    %构造离散阶跃输入信号
u(k) = kp * x(1) + kd * x(2) + ki * x(3);          %PID 控制器的实现
if u(k)>= 10
   u(k) = 10;
end
if u(k)<= -10
   u(k) = -10;
end                                                %控制器的输出限幅
%Linear model
yout(k) = -den(2) * y_1 - den(3) * y_2 - den(4) * y_3 + num(2) * u_1 + num(3) * u_2 + num(4) * u_3;
error(k) = rin(k) - yout(k);
%Return of parameters
u_3 = u_2;u_2 = u_1;u_1 = u(k);
y_3 = y_2;y_2 = y_1;y_1 = yout(k);
x(1) = error(k);                                   %Calculating P
x(2) = (error(k) - error_1)/ts;                    %Calculating D
x(3) = x(3) + error(k) * ts;                       %Calculating I
error_1 = error(k);
end
figure(1);
plot(time,rin,'k',time,yout,'k');
xlabel('时间'),ylabel('输入输出 PID 跟踪效果');
```

其仿真的跟踪结果如图 8.5 所示。

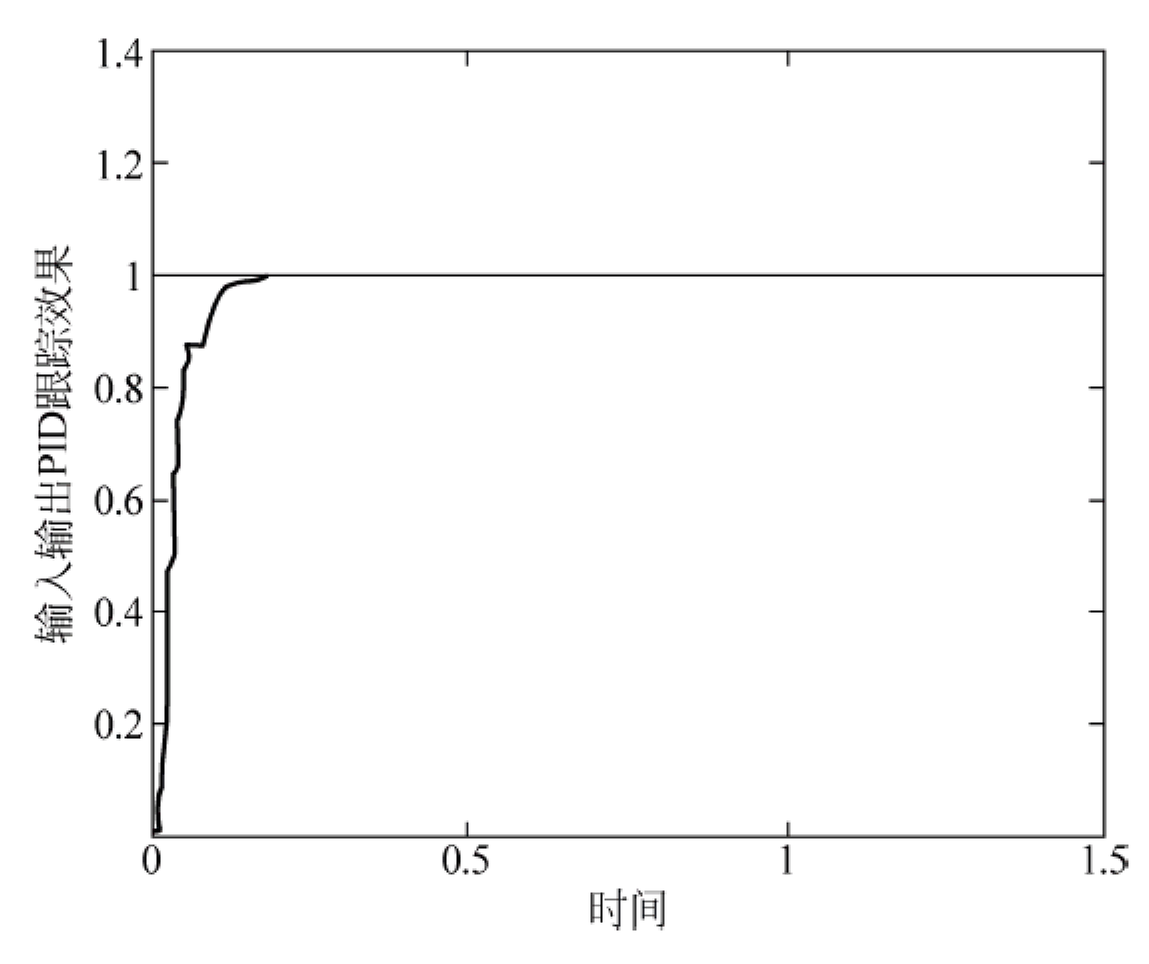


图 8.5　PID 阶跃跟踪效果图

在实际的控制系统中，PID 类的控制器占 90% 以上，通过例 8.1 的介绍，借助于 MATLAB 语言实现了 PID 控制器的设计，其他算法的编程可以参考有关资料。

8.2.3　显示驱动技术

在计算机控制系统中，显示系统是其重要的组成部分，是人机对话的重要途径，主要用来显示生产过程的工艺状况与运行结果，以便现场工作人员的正确操作。常用的显示器件有指示灯方式、显示记录仪、阴极射线管(Cathode Ray Tube，CRT)、发光二极管显示器(Light Emitting Diode，LED)、液晶显示器(Liquid Crystal Display，LCD)，以及最近发展起来的新型薄膜晶体管(Thin Film Transistor，TFT)方式等。

常用的图形显示器主要有平面显示器和 CRT 显示器两种。CRT 显示器不仅可以显示字符，还可以进行各种画面以及各种工况流程的显示，和计算机配合使用可以图文并茂十分方便地实现生产过程的集中管理和监视，是目前在工业控制计算机系统中应用最多，也是技术最成熟的图形显示技术，一般适用于大型的计算机控制系统，特别在 DDC、SCC 以及 DCS 系统中，大都采用 CRT 操作台进行监视和控制。

对于一些中小型的数据采集和控制监视系统，常选用 LED 和 LCD 作为其显示器件。LED 和 LCD 都具有结构简单，体积小，功耗低，配置灵活，响应速度快，显示清晰，可靠性高和寿命长的优点。LCD 是一种新型的显示器件，它利用液晶分子受电场作用而影响照射其上面的光线的散射方向，易于形成各种图形和数字的原理制成的。LCD 可分为字符型、段位型和点阵型三种，它具有功耗极低的优点，多用于仪表系统和低功耗应用系统中。

LED 因其成本低廉，与单片机接口方便等特点，在小型计算机控制系统中得到了极为广泛的应用，这些系统中往往只需几个简单的数字显示或字符状态便可满足现场的需求。这里重点介绍 LED 的显示驱动原理。

1. LED 数码管结构

LED 显示器件是通过发光二极管显示字段的器件，LED 显示器有多种类型，单段的圆形或方形 LED 常用来显示设备的运行状态；16 段的 LED 呈现“米”字形，可以显示各种数

字和字符；通常使用的是 7 段 LED 数码管，这 7 段构成一个“日”字形，通过点亮不同的字段来表示不同的数字或者字符，有的还外加一段用于表示小数点，8 段数码管的外形结构如图 8.6(a)所示。这样，通过不同的组合可用来显示数字 0～9、字符 A～F、H、L、P、R、U、Y、符号“-”及小数点“.”。多个 8 段显示器就可以实现多个数字或字符的组合，颜色也有红、黄、绿、白等几种颜色来供选择。

常用的 8 段 LED 数码管又分为共阴极和共阳极两种结构，分别如图 8.6(b)和图 8.6(c)所示。

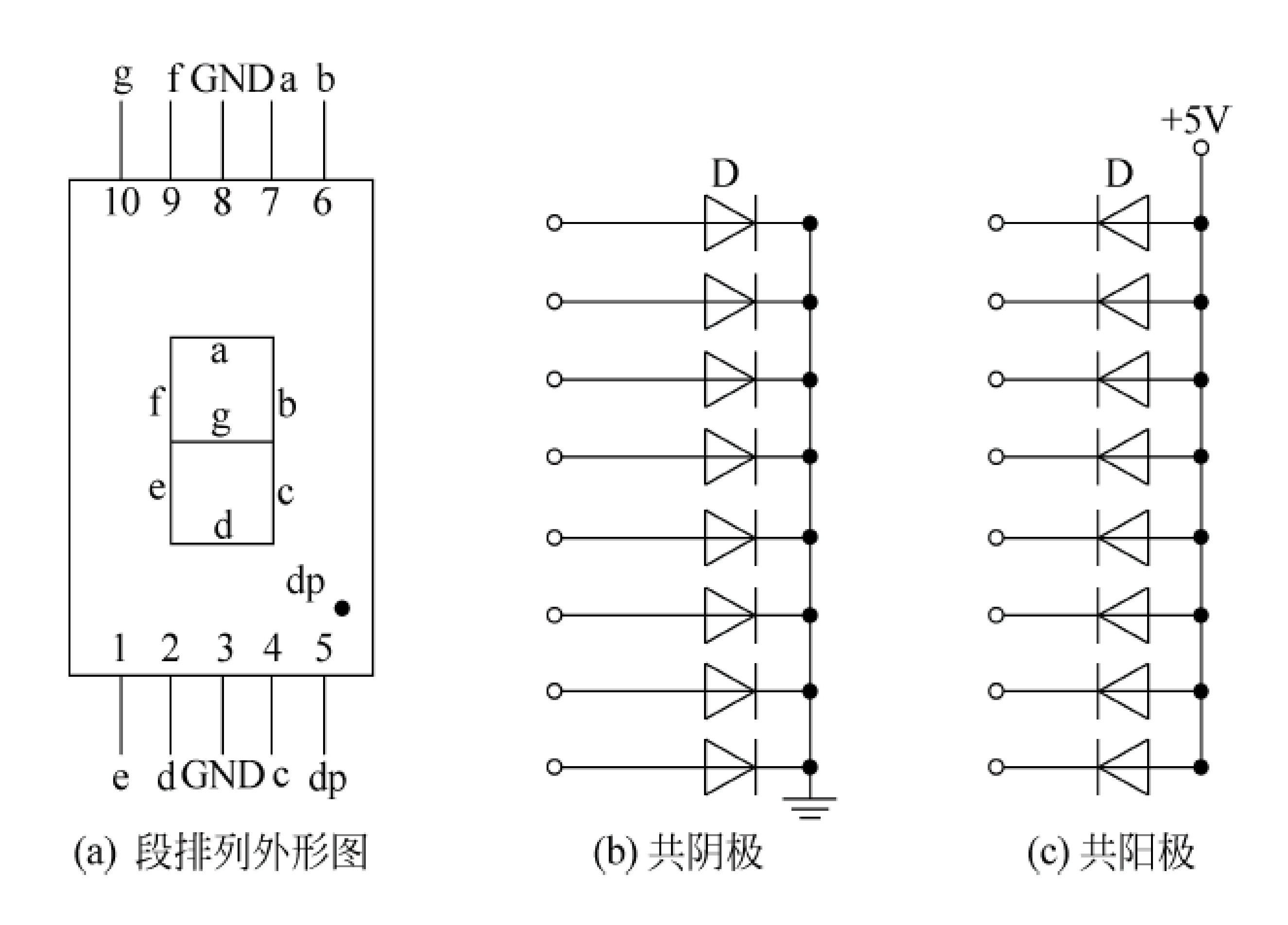


图 8.6　8 段 LED 数码管结构图

2. LED 数码管工作原理

共阴极数码管的 8 个发光二极管的阴极(二极管负端)连接在一起，通常，公共阴极接低电平(一般接地)，其他管脚接段驱动电路输出端，当某段驱动电路的输出端为高电平时，则该端所连接的字段导通并点亮，根据发光字段的不同组合可显示出各种数字或字符。此时，要求段驱动电路能提供额定的段导通电流，还需根据外接电源及额定段导通电流来确定相应的限流电阻。

共阳极数码管的 8 个发光二极管的阳极(二极管正端)连接在一起，通常，公共阳极接高电平(一般接电源)，其他管脚接段驱动电路输出端。当某段驱动电路的输出端为低电平时，则该端所连接的字段导通并点亮，根据发光字段的不同组合可显示出各种数字或字符。此时，要求段驱动电路能吸收额定的段导通电流，还需根据外接电源及额定段导通电流来确定相应的限流电阻。

3. 数码管字形编码

通常把控制 LED 数码管发光显示的 8 位字节数据称为段选线、字符译码或字模，要使数码管显示出相应的数字或字符必须使段数据口输出相应的字形编码。假设当段引脚 dp～a 与 CPU 数据位 D7～D0 一一对应相连接时，也即如下的对应关系。

LED 段：dp　g　f　e　d　c　b　a

数据线：D7　D6　D5　D4　D3　D2　D1　D0

如使用共阳极数码管，数据为 0 表示对应字段亮，数据为 1 表示对应字段暗；如使用共

阴极数码管，数据为 0 表示对应字段暗，数据为 1 表示对应字段亮。如要显示“0”，共阳极数码管的字形编码应为：11000000B(即 C0H)；共阴极数码管的字形编码应为：00111111B(即 3FH)。通过不同的段选码，即可显示出不同的相应字符，共阴极和共阳极 LED 数码管字形编码如表 8.1 所示。

表 8.1 LED 数码管字模表

显示字符	字形	共阳极									共阴极								
		dp	g	f	e	d	c	b	a	字形码	dp	g	f	e	d	c	b	a	字形码
0	0	1	1	0	0	0	0	0	0	C0H	0	0	1	1	1	1	1	1	3FH
1	1	1	1	1	1	1	0	0	1	F9H	0	0	0	0	0	1	1	0	06H
2	2	1	0	1	0	0	1	0	0	A4H	0	1	0	1	1	0	1	1	5BH
3	3	1	0	1	1	0	0	0	0	B0H	0	1	0	0	1	1	1	1	4FH
4	4	1	0	0	1	1	0	0	1	99H	0	1	1	0	0	1	1	0	66H
5	5	1	0	0	1	0	0	1	0	92H	0	1	1	0	1	1	0	1	6DH
6	6	1	0	0	0	0	0	1	0	82H	0	1	1	1	1	1	0	1	7DH
7	7	1	1	1	1	1	0	0	0	F8H	0	0	0	0	0	1	1	1	07H
8	8	1	0	0	0	0	0	0	0	80H	0	1	1	1	1	1	1	1	7FH
9	9	1	0	0	1	0	0	0	0	90H	0	1	1	0	1	1	1	1	6FH
A	A	1	0	0	0	1	0	0	0	88H	0	1	1	1	0	1	1	1	77H
B	B	1	0	0	0	0	0	1	1	83H	0	1	1	1	1	1	0	0	7CH
C	C	1	1	0	0	0	1	1	0	C6H	0	0	1	1	1	0	0	1	39H
D	D	1	0	1	0	0	0	0	1	A1H	0	1	0	1	1	1	1	0	5EH
E	E	1	0	0	0	0	1	1	0	86H	0	1	1	1	1	0	0	1	79H
F	F	1	0	0	0	1	1	1	0	8EH	0	1	1	1	0	0	0	1	71H
H	H	1	0	0	0	1	0	0	1	89H	0	1	1	1	0	1	1	0	76H
L	L	1	1	0	0	0	1	1	1	C7H	0	0	1	1	1	0	0	0	38H
P	P	1	0	0	0	1	1	0	0	8CH	0	1	1	1	0	0	1	1	73H
U	U	1	1	0	0	0	0	0	1	C1H	0	0	1	1	1	1	1	0	3EH
-	-	1	0	1	1	1	1	1	1	BFH	0	1	0	0	0	0	0	0	40H
.	.	0	1	1	1	1	1	1	1	7FH	1	0	0	0	0	0	0	0	80H
灭	灭	1	1	1	1	1	1	1	1	FFH	0	0	0	0	0	0	0	0	00H

4. LED 显示器的显示方式

在计算机控制系统中，常利用 n 个 LED 显示器来构成 n 位显示，通常把点亮 LED 某一段的控制称为段选，而把点亮 LED 某一位的控制称为位选或片选。根据 LED 显示器的段选线、位选线与控制端口的连接方式不同，LED 显示器有静态显示与动态显示两种方式。

静态显示是指数码管显示某一字符时，相应的发光二极管恒定导通或恒定截止。这种显示方式的各位数码管相互独立，公共端恒定接地(当共阴极时)或接正电源(当共阳极时)。每个数码管的 8 个字段分别与一个 8 位 I/O 口地址相连，I/O 口只要有段码值输出，相应的字符即可显示出来，并保持不变，直到 I/O 口有新的段码来输出刷新。采用静态显示方式，较小的电流即可获得较高的亮度，且占用 CPU 时间少，编程简单，其显示方式便于监测和控制，但其占用的口线多，硬件电路比较复杂，成本高，只适合于显示位数较少的场合。

动态显示是一位一位地轮流点亮各位数码管，这种逐位点亮显示器的方式称为位扫描。通常，各位数码管的段选线相应并联在一起，由一个8位的I/O口控制，各位的位选线（公共阴极或阳极）由另外的I/O口线控制。动态方式显示时，各数码管分时轮流选通，要使其稳定显示必须采用扫描方式，即在某一时刻只选通一位数码管，并送出相应的段码，在另一时刻选通另一位数码管，并送出相应的段码，依此规律循环，即可使各位数码管显示将要显示的字符，虽然这些字符是在不同的时刻分别显示的，但由于人眼存在视觉暂留效应，只要每位显示间隔时间足够短就可以给人以同时显示的感觉。

采用动态显示方式比较节省I/O口，硬件电路也比静态显示方式简单，但其亮度不如静态显示方式，而且在显示位数较多时，CPU要依次扫描，占用CPU较多的时间。

LED显示器显示方式的接口电路和程序流程图不再详细讲解，读者可以参考有关资料。

8.3 OPC技术

随着计算机技术的发展，计算机在工业控制领域发挥着越来越重要的作用。各种控制设备之间及控制设备与计算机之间的信息交换是通过驱动程序也即相应的通信协议来实现的，但是不同厂家生产的产品其协议互不相同，即使同一厂家的不同设备与计算机之间的驱动程序也是不同的。因此为了访问现场设备的数据信息，应用软件开发商都需要为不同的设备编写大量的驱动程序，而计算机硬件厂家也要为不同的应用软件编写不同的驱动程序。由于现场设备的种类繁多，且产品的不断升级，往往给用户和软件开发商带来了巨大的工作负担。通常这样也不能满足工作的实际需要，程序的可复用程度低，也不符合软件工程的发展趋势。因此，系统集成商和开发商急切需要一种具有高效性、可靠性、开放性、可互操作性的即插即用的设备驱动程序，OPC技术就是在这种情况下应运而生，它为工业控制现场不同设备之间的通信提供了统一的接口规范。

OPC把OLE应用于工业控制领域。OLE原意是对象连接和嵌入，但随着OLE 2的发行，其范围已远远超出了这个概念。OPC建立于OLE规范之上，它为工业控制领域提供了一种标准的数据访问机制。现在的OLE包含了许多新的特征，比如统一数据传输、结构化存储和自动化，已经成为独立于计算机语言、操作系统，甚至硬件平台的一种规范，是面向对象程序设计概念的进一步推广。

OPC规范是在微软的倡导下，世界上多个自动化公司、软硬件供应商与微软合作开发的一套工业标准，是专为在现场设备、自控应用、企业管理应用软件之间实现系统无缝集成而设计的硬件和软件接口规范。它基于微软现有的OLE、组件对象模型COM（Component Object Model）、分布式组件对象模型DCOM（Distributed COM）技术，定义了一套适于过程控制应用，支持过程数据访问、报警、事件与历史数据访问等功能的接口，便于不同供应商的软硬件实现“即插即用”的连接与系统集成，目前已得到越来越多的工控领域硬件和软件制造商的承认和支持，并成为事实上的国际标准。

1. OPC规范

OPC规范采用客户/服务器模型，包括OPC服务器和OPC客户端两个部分，其实质是

在硬件供应商和软件开发商之间建立了一套完整的“规则”，只要遵循这套规则，数据交互对两者而言都是透明的，硬件供应商无需考虑应用程序的多种需求和传输协议，只需要提供一套符合 OPC Server 规范的程序组，便能够提供一个功能齐备的应用接口，而软件开发商也无需了解硬件的实质和操作过程，只需要一套具备 OPC 客户能力的软件，就可以与所有符合 OPC 服务器规范的程序组连接，来获取需要的数据。

OPC 技术规范是由 OPC 基金会制定的，它是所有支持 OPC 技术的 Server 和 Client 开发者都要遵循的技术规范。此规范详细地定义了在实现 Server 对象和 Client 对象的开发过程中，开发者要支持的接口及接口中所包含的方法。OPC 技术包括的几种规范，有数据存取(Data Access)规范、报警与事件(Alarm and Event)规范、历史数据存取(Historical Data Access)规范、安全(Security)规范和批处理(Batch)规范。

2. OPC 的特点

OPC 是为了解决应用软件与各种设备驱动程序的通信而产生的一项工业技术规范和标准。它采用客户/服务器体系，基于 Microsoft 的 OLE/COM 技术，为硬件厂商和应用软件开发者提供了一套标准的接口。

综合起来说，OPC 有以下几个特点。

(1) 计算机硬件厂商只需要编写一套驱动程序就可以满足不同用户的需要。硬件供应商只需提供一套符合 OPC Server 规范的程序组，无需考虑工程人员需求。

(2) 应用程序开发者只需编写一个接口便可以连接不同的设备。软件开发商无需重新编写大量的设备驱动程序。

(3) 工程人员在设备选型上有了更多的选择。对于最终用户而言，选择面更宽一些，可以根据实际情况的不同，选择切合实际的设备。

OPC 扩展了设备的概念，只要符合 OPC 服务器的规范，OPC 客户都可与之进行数据交换，而无需了解设备究竟是 PLC 还是仪表。甚至只要在数据库系统上建立了 OPC 规范，OPC 客户便可与之方便地实现数据交互。

总的来说，OPC 是为解决应用软件与各种设备驱动程序的通信而产生的一个工业技术标准，它把硬件厂商和应用软件开发者分离开来，使得双方的工作效率都有了很大的提高，因此在短时间内 OPC 取得了飞速的发展。

3. OPC 的适用范围

OPC 的设计者们最终目标是在工业领域建立一套数据传输规范，并为之制定了一系列的发展计划。现有的 OPC 规范涉及如下领域。

(1) 在线数据监测。OPC 实现了应用程序和工业控制设备之间高效、灵活的数据读写。

(2) 历史数据访问。OPC 实现了读取、操作、编辑历史数据库的方法。

(3) 远程数据访问。可借助 Microsoft 的 DCOM 技术，OPC 实现了高性能的远程数据访问能力。

(4) 报警和事件处理。OPC 提供了 OPC 服务器发生异常时，以及 OPC 服务器设定事件到来时向 OPC 客户发送通知的一种机制。

(5) OPC 实现的功能还包括安全性、批处理、历史报警事件数据访问等。

4. OPC 的基本概念

OPC 服务器由服务器(Server)、组(Group)、数据项(Item)三类对象组成,相当于三种层次上的接口。OPC 过程控制系统结构采用客户/服务器模式,通常把符合 OPC 规范的设备驱动程序称为 OPC 服务器,而将符合 OPC 规范的应用软件统称为 OPC 客户。服务器充当客户与硬件设备之间的桥梁,客户对硬件设备的数据读写操作由服务器代理完成,客户不需要同硬件设备直接打交道,或者说客户是独立于设备的(即不管现场设备以何种形式存在,客户都以统一的方式去访问)。

(1) 服务器对象(Server)拥有服务器的所有信息,同时也是组对象(Group)的容器。组对象(Group)拥有本组的所有信息,同时包容并逻辑组织 OPC 数据项(Item)。

(2) OPC 组对象(Group)包含本组的所有信息,同时包含并管理 OPC 数据项。组对象提供了客户组织数据的一种方法。组是应用程序组织数据的一个单位,客户可对之进行读写,还可设置客户端的数据更新速率。当服务器缓冲区内数据发生改变时,OPC Server 将向客户发出通知,客户得到通知后再进行必要的处理,而无需浪费大量的时间进行查询。OPC 规范定义了两种组对象:公共组(或称全局组,Public)和局部组(私有组,Local)。公共组由多个客户共有,私有组只隶属于一个 OPC 客户。全局组对所有连接在服务器上的应用程序都有效,而私有组只对与其建立连接的 Client 有效。一般说来,客户和服务器的一对连接只需要定义一个组对象。在每个组对象中,客户可以加入多个 OPC 数据项(Item)。

(3) OPC 数据项是服务器端定义的对象,也是读写数据的最小逻辑单位,一个数据项与一个具体的位号相连,通常指向设备的一个寄存器单元,存储具体 Item 的定义、数据值、状态值等信息。OPC 客户对设备寄存器的操作都是通过其数据项来完成的。通过定义数据项,OPC 规范尽可能的隐藏了设备的特殊信息,也使 OPC 服务器的通用性大大增强。OPC 数据项并不提供对外接口,不能独立于组存在,必须隶属于某一个组,客户不能直接对之进行操作,所有操作都是通过组对象进行的。

OPC 客户端是访问过程数据、消息和 OPC 服务器归档的应用程序,而 OPC 服务器则是处理过程数据的应用程序、各种网络协议和访问这些数据的接口之间的中间设备,它提供给不同制造商的应用程序一个标准的软件接口,在客户端和服务器端各自定义了统一的标准接口,接口具有不变特性,OPC 所提供的接口标准事实上是一种“软件接口标准”或“软件总线”,它明确定义了客户同服务器之间的通信机制,是连接客户同服务器的桥梁和纽带。应用程序作为 OPC 接口中的 Client 方,硬件驱动程序作为 OPC 接口中的 Server 方,每个 OPC Client 应用程序都可以连接若干个 OPC Server,每一个硬件驱动程序也可以为若干个应用程序提供数据。图 8.7 给出了 OPC 的体系结构。

这样,就可以自由选择最符合实际控制要求的软、硬件产品,然后将它们像“搭积木”一样组合在一起进行无缝地工作,这一切都来源于 OPC 所提供的强大互操作性的特点,对此我们可以作一个形象的比喻:尽管家用电器的种类繁多,但它们的电源插头与插座却是统一的。可以这样说,统一的标准接口是 OPC 的实质和灵魂。

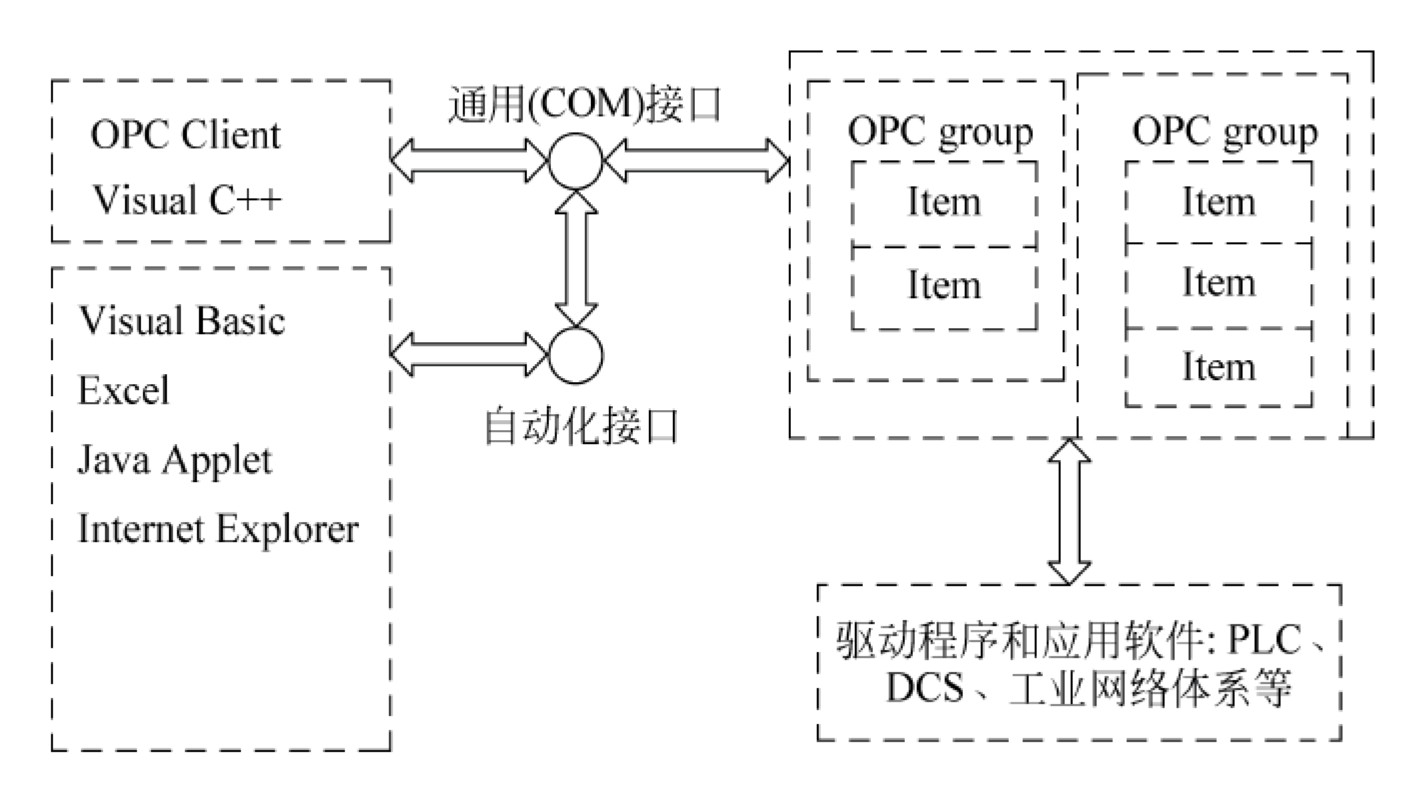


图 8.7 OPC 体系结构图

5. 客户操作 OPC 数据项的一般步骤

读写 OPC 数据项的一般步骤如下。

(1) 通过服务器对象接口枚举服务器端定义的所有数据项，如果客户对服务器所定义的数据项非常熟悉，此步可以忽略。

(2) 将要操作的数据项加入客户定义的组对象中。

(3) 通过组对象对数据项进行读写等操作。

每个数据项的数据结构包括三个成员变量，即数据值、数据质量和时间戳。数据值是以 Variant 形式表示的。应当注意，数据项表示同数据源的连接而不等同于数据源，无论客户是否定义数据项，数据源都是客观存在的，可以把数据项看作数据源的地址，即数据源的引用，而不应看作数据源本身。

6. OPC 的报警(Alarm)和事件(Event)

报警和事件处理机制增强了 OPC 客户处理异常的能力。服务器在工作过程中可能出现异常，此时，OPC 客户可通过报警和事件处理接口得到通知，并能通过该接口获得服务器的当前状态。在很多场合，报警(Alarm)和事件(Event)的含义并不加以区分，两者也经常互换使用。从严格意义上讲，两者含义略有差别。

依据 OPC 规范，报警是一种异常状态，是 OPC 服务器或服务器的一个对象可能出现的所有状态中的一种特殊情况。例如，服务器上标记为 FC101 的一个单元可能有如下状态：高出警戒，严重高出警戒，正常，低于警戒，严重低于警戒。除了正常状态外，其他状态都视为报警状态。

事件则是一种可以检测到的出现的情况，这种情况或来自 OPC 客户，或来自 OPC 服务器，也可能来自 OPC 服务器所代表的设备，通常都有一定的物理意义。事件可能与服务器或服务器的一个对象的状态有关，也可能毫无关系。如高出警戒和正常状态的转换事件和服务器的某个对象的状态有关，而操作设备，改变系统配置以及出现系统错误等事件和对象状态就无任何关系。

7. OPC 的接口方式

OPC 规范提供了两套接口方案，即 COM 接口和自动化。COM 接口效率高，通过该接

口，客户能够发挥 OPC 服务器的最佳性能，采用 C++语言的客户一般采用 COM 接口方案；自动化接口使解释性语言和宏语言访问 OPC 服务器成为可能，采用 VB 语言的客户一般采用自动化接口。自动化接口使解释性语言和宏语言编写客户应用程序变得简单，然而自动化客户运行时需进行类型检查，这一点则大大牺牲了程序的运行速度。

OPC 服务器必须实现 COM 接口，是否实现自动化接口则取决于供应商的主观意愿。COM 是位于同一台计算机上对象之间通信的标准协议。其中，对象是指属于不同程序的一部分，服务器是提供服务的对象，客户机是使用服务器提供服务的应用程序。当 OPC 服务器与客户机运行在同有一台电脑上的时候，客户机与服务器通过本机的 COM 库进行数据的传递。DCOM 代表 COM 功能的扩展，从而允许对远程网络计算机上的对象进行访问。该机制允许在工业、管理办公室和生产的应用程序之间进行标准化的数据交换。当客户机与服务器运行在不同的网络节点上时，服务器与客户程序之间通过 DCOM 方式进行通信，也即通过网络可以访问其他计算机上的 OPC 服务器，从而达到远程数据连接的目的。

8. OPC 的数据访问方式

1）服务器缓冲区数据和设备数据

OPC 服务器本身就是一个可执行程序，该程序以设定的速率不断地同物理设备进行数据交互。服务器内有一个数据缓冲区，其中存有最新的数据值、数据质量戳和时间戳。时间戳表明服务器最近一次从设备读取数据的时间。服务器对设备寄存器的读取是不断进行的，时间戳也在不断更新。即使数据值和质量戳都没有发生变化，时间戳也会进行更新。客户既可从服务器缓冲区读取数据，也可直接从设备读取数据，从设备直接读取数据速度会慢一些，一般只有在故障诊断或极特殊的情况下才会采用。

2）同步和异步

OPC 客户和 OPC 服务器进行数据交互可以有两种不同方式，即同步方式和异步方式。同步方式实现较为简单，当客户数目较少而且同服务器交互的数据量也比较少的时候可以采用这种方式；异步方式实现较为复杂，需要在客户程序中实现服务器回调函数。然而当有大量客户和大量数据交互时，异步方式能提供高效的性能，尽量避免阻塞客户数据请求，并最大可能地节省 CPU 和网络资源。

8.4　数据处理技术

在计算机控制系统中，需要对生产过程中的各种信号进行测量，因此数据采集处理是其最基本的功能模式之一。测量时，一般是通过传感器或变送器把生产过程中的信号转换成电信号，然后经过 A/D 或 DO 通道将相应的数字量送入计算机中，计算机在对这些数字量进行控制和显示之前，还必须进行相应的数据处理。

计算机控制技术中数据处理一般从三个方面来说，一是对由传感器输出的信号进行信号的放大、滤波以及隔离等处理，这也就是常说的信号调理，这部分主要由硬件电路来完成；二是对采集到计算机中的信号数据进行一些处理，比如数据预处理、数字滤波、非线性处理、标度变换等处理，这也是常说的一次处理；三是对一次处理后得到的数据进行分析、判断，由控制器生成所需的控制信号，这也就是所谓的二次处理。一次处理和二次处理主要是由

软件来实现的,本节所说的数据处理主要是指一次处理。经过一次处理可以使数据格式化、标准化,以便符合进一步处理的需要。

8.4.1 数据预处理技术

对测量数据的预处理是计算机控制系统数据处理的基础,主要包括数字调零、系统误差的校正以及输出数据的极性与字长的预处理。这里主要介绍系统误差的校正方法。

系统误差是指在相同条件下,经过多次测量,误差的数值(包括大小和符号)保持恒定,或者按某种已知的规律变化的误差。它的特点是在一定的测量条件下,其变化规律是可以掌握的,产生误差的原因一般也是知道的。因此,系统误差原则上是可以通过适当的技术途径来确定并加以校正的。在控制系统的测量输入通道中,一般均存在放大器等器件的零点偏移和漂移,会造出放大电路的增益误差和器件参数的不稳定等现象,它们都会影响到系统测量数据的准确性,这些误差都属于系统误差。有时必须对这些系统误差进行校正,一般采用软件程序进行处理,即可实现对这些系统误差进行自动校准。

1. 数字调零

零点偏移是造出系统误差的主要原因之一,常把采用程序来实现零点调整的方法称为数字调零。数字调零电路如图 8.8 所示。

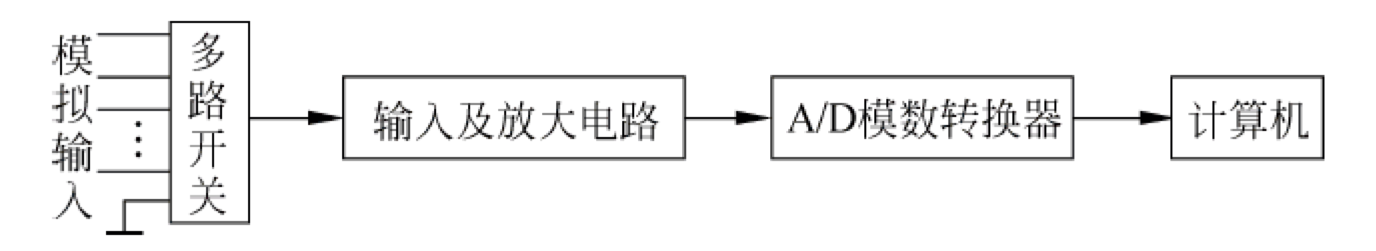


图 8.8 数字调零电路

在测量时,首先将多路开关的输入接地,理论上电压为零的信号,经过放大电路、A/D 转换电路进入到计算机中的数值也应为零,而实际上由于零点偏移产生了一个不等于零的数值 x_0,这个值就是零点偏移值,并将测得的数据存储起来。正常测量时,依次采集每一路输入电压进行测量,测出此时的输入值为 x,用每次测量结果 x 减去 x_0 即为此次测量的实际输入值。很显然,采用这种方法,可去掉输入电路、放大电路、A/D 转换电路本身的偏移以及随时间和温度而发生的各种漂移的影响,从而大大降低对这些电路器件的偏移值的要求,降低了系统的硬件成本。

2. 系统校准

在数字调零校正中不能消除由传感器本身引入的误差,为了克服这种缺点,可采用系统校准处理技术。系统校准的原理与数字调零差不多,只是把测量电路扩展到了现场底层的传感器,而且不是在每次采集数据时都进行校准,而是由人工在需要时接入标准的参数进行校准测量,并把测得的数据存储起来,供以后实际测量时使用。一般自动校准系统只测一个标准输入信号 y_R,零点漂移的补偿仍由数字调零来完成。

设数字调零后测出接标准输入信号 y_R 时的数据为 x_R,而实际测得输入信号为 y 时的数据为 x,则可按式(8.1)来计算 y。

$$y=\frac{y_R}{x_R}x \tag{8.1}$$

系统校准特别适合于传感器特性随时间会发生变化的场合。比如常用的电容式湿度传感器,其输入输出的特性会随着时间的变化而发生变化,一般一年以上变化会大于其精度允许值,这时可根据需要每隔一段时间,用其他精确方法测出这时的湿度值,然后把它作为校准值输入测量系统,在以后实际测量湿度时,计算机将自动用该输入值来校准以后的测量值。

8.4.2 数字滤波技术

由于工业控制对象的环境一般比较恶劣,干扰源比较多,比如环境温度、电磁场和噪声等,虽然在硬件上可以接入一个 *RC* 低通滤波器来抑制工频及其以上频率的干扰,但对于频率很低的干扰信号,用 *RC* 滤波器就很难达到抑制的目的。因此经过硬件电路的滤波处理,仍会混有随机干扰噪声,从而使采样值偏离实际值。如果仅仅采样一次,则无法确定该结果是否可信,为了减少干扰噪声等对采样值的影响,提高系统的可靠性, 在进行数据处理和PID 调节之前,一般先对采样值进行数字滤波。

所谓数字滤波,其实质是一种程序滤波,就是计算机系统对输入信号进行多次采样,然后通过一定的计算机程序对采样信号进行平滑加工,去掉原始数据中掺杂的噪声数据,提高其有用信号的比重,消除和减少各种干扰和噪声,以保证计算机系统的可靠性和稳定性。

数字滤波克服了模拟 *RC* 滤波器的不足,与模拟滤波器相比有以下几个优点。

(1) 由于数字滤波是用程序来实现的,多个输入通道可以共用一个滤波程序,改变滤波方法只需改变滤波程序,很容易实现,同时也不需要增加硬件设备,因而系统的可靠性高,稳定性好,各回路之间不存在阻抗的匹配问题。

(2) 数字滤波可以对频率很低的信号进行滤波,而模拟滤波器由于受电容电量的影响,在低频段滤波难以取得较好的滤波效果。

(3) 通过改变数字滤波程序,就可以根据需要选择不同的滤波方法和调整滤波参数,这比修改模拟滤波器的硬件灵活、方便。

总之,数字滤波与硬件滤波器相比优点比较多,因此在计算机控制系统中得到了广泛的应用,数字滤波的方法很多,不同的方法各有不同的特点,常用的有中值滤波、程序滤波、平均值滤波、惯性滤波等,下面主要介绍常用的几种数字滤波方法。

1. 中值滤波

所谓中值滤波就是对某一目标参数连续采样 N 次(一般 N 取奇数),然后将这些采样值按从小到大或者从大到小进行排序,再取其中间值作为本次有效的采样值。

中值滤波对于偶然因素引起的波动或采样器不稳定造成的误差所引起的脉动干扰比较有效,若变量变化缓慢,采用中值滤波效果比较良好,但对于快速变化的过程参数,比如流量,则不宜采用。

在实际使用时,N 值的选取要适当。N 过小可能起不到去除干扰的作用; N 过大,会造成采样数据的时延过大,造成系统的性能变差,一般取 3～5 次即可。在编制中值滤波算法程序时,可采用几种常规的排序算法比如冒泡法将采样值进行排队,然后再取其中间值

即可。

2. 程序判断滤波

当采样信号由于随机干扰和误检测或者变送器不稳定而引起严重失真时,可以采用程序判断滤波。程序判断滤波是基于对具体控制对象的实际情况分析的基础上进行的,它根据实际生产经验,确定出相邻两次采样信号之间可能出现的最大偏差 ΔY,若超过此偏差值,则表明该输入信号是干扰信号,应该去掉;若小于或等于此偏差值,可将该信号作为本次采样值。

程序滤波根据滤波方法的不同,可分为限幅滤波和限速滤波两种。

1) 限幅滤波

限幅滤波就是把两次相邻的采样值进行相减,求出其增量的绝对值,然后再与两次采样所允许的最大偏差(由被控对象的实际情况决定)ΔY 进行比较,如果小于或等于 ΔY,则取本次采样值;如果大于 ΔY,则仍取上次采样值作为本次采样值。其算法如下。

若 $|Y(k)-Y(k-1)|\leqslant\Delta Y$,则 $Y(k)=Y(k)$,取本次采样值。

若 $|Y(k)-Y(k-1)|>\Delta Y$,则 $Y(k)=Y(k-1)$,取上次采样值。

式中,$Y(k)$为第 k 次采样;$Y(k-1)$为第 $k-1$ 次采样;ΔY 为两次相邻采样值所允许的最大偏差,其大小取决于采样周期 T 及 Y 值的变化动态响应。

限幅滤波使用时的关键问题是最大允许误差 ΔY 的选取,通常 ΔY 可根据经验数据来选取,必要时可由实验得出。ΔY 太大,各种干扰信号将“乘虚而入”,使系统误差增大;ΔY 太小,又会将一些有用信号“拒之门外”,使计算机采样效率变低。限幅滤波主要用于变化比较缓慢的参数,如温度、物位等测量系统。

2) 限速滤波

限速滤波同样是以本次采样值减去上次的采样值求出其增量的绝对值来与设定的最大允许误差 ΔY 进行比较,如果小于或等于 ΔY,则取 $Y(k)$为本次采样值;否则再采样一次把所采得值 $Y(k+1)$与 $Y(k)$进行比较,若$|Y(k+1)-Y(k)|\leqslant\Delta Y$,则以 $Y(k+1)$作为本次采样值;否则就以$[Y(k+1)+Y(k)]/2$ 的值作为本次采样值。

限幅滤波是用两次采样值来决定采样的结果,而限速滤波有时需要三次采样值来决定采样结果。假设顺序采样时刻 t1、t2、t3 所采集到的数据分别为 $Y(1)$、$Y(2)$、$Y(3)$,则

当$|Y(2)-Y(1)|\leqslant\Delta Y$ 时,采用 $Y(2)$作为本次采样值。

当$|Y(2)-Y(1)|>\Delta Y$ 时,不采用 $Y(2)$,但保留,继续采样取得 $Y(3)$。

当$|Y(3)-Y(2)|\leqslant\Delta Y$ 时,采用 $Y(3)$作为本次采样值。

当$|Y(3)-Y(2)|>\Delta Y$ 时,则取$(Y(3)+Y(2))/2$ 为采样值。

限速滤波是一种折中的方法,它既照顾了采样的实时性,又顾及了采样值变化的连续性。限速滤波的缺点是不够灵活,以及不能反映采样点数大于 3 时各采样数值受到的干扰情况,因此在实际应用中受到限制。

3. 平均值滤波

平均值滤波就是对多个采样值进行平均算法,这是消除随机误差最常用的方法之一。具体又可分为如下几种。

1）算术平均滤波

算术平均滤波就是对被测参数连续采样 N 次，把 N 个采样值相加后的算术平均值作为本次的有效采样值。这种方法可以减少系统的随机干扰对采样结果的影响，其算法为

$$Y(k) = \frac{1}{N}\sum_{i=1}^{N} X(i) \tag{8.2}$$

式(8.2)中，N 为采样次数；X_i 为第 i 次的采样值。采样次数 N 决定了信号的平滑度和灵敏度。显然，N 越大，平滑度越高，但是系统的灵敏度随之降低，所以采样次数 N 的取值视被控对象的具体情况而定。一般情况下，流量信号可取 12 左右，压力信号可取 4 左右，温度、成分等缓慢变化的信号可取 2 甚至无噪声时可不进行算术平均。在编制算法程序时，N 的值一般取 2、4、8、16 之类的 2 的整数幂，这样以便于在程序中使用移位来代替除法求得平均值。

算术平均滤波法主要用于对压力、流量等周期脉动信号的采样值进行平滑处理，但对脉冲性干扰的平滑不够理想，因此它不适用脉冲性干扰比较严重的场合。

2）加权平均滤波

算术平均滤波算法中，对于所用的 N 个采样值来说，所占的比例是相同的，滤波的结果取每个采样值的 $1/N$。有时为了提高滤波效果，将各采样值取不同的比例，然后再相加，此方法称为加权平均法。具有 N 个采样值的加权平均值公式为

$$Y(k) = \sum_{i=1}^{N} C_i X(i) \tag{8.3}$$

式(8.3)中，C_1、C_2、C_3、…、C_N 均为常数，称为各采样值的系数，应满足以下关系：

$$\sum_{i=1}^{N} C_i = 1 \tag{8.4}$$

C_i 体现了各采样在平均值中所占的比例，可以根据具体情况决定。对于正在变化的信号，如采集流量的值，一般采样次数愈靠后，取的比例愈大，这样可以增加新的采样值在平均值中的比例。这种滤波方法可以根据需要突出信号中的某一部分，或抑制信号中的某一部分。

3）滑动平均滤波

前面介绍的不管是算术平均滤波还是加权平均滤波，每取得一个有效采样值都需要连续采样 N 个数据，然后求算术平均值或加权平均值，这种方法适合于有脉动式干扰的场合。但由于采样 N 个数据需要的时间比较长，故检测速度较慢。当系统的采样速度较慢或采样信号变化较快时，系统的实时性就很难保证。而滑动平均值滤波可克服此缺点，滑动平均滤波是在每个采样周期只采样一次，将这一次的采样值和过去的若干采样值一起求平均，所得的结果即为有效采样值。

滑动平均滤波的具体做法是先在 RAM 中建立一数据缓冲区，依次存放 N 个采样数据，把这 N 个采样数据看成一个队列，队列的长度固定为 N，然后每采进一个新数据，就把采样结果放入队尾，而将最早采集的队首的那个数据丢掉，这样在队列中始终有 N 个“最新”的数据，然后求出包括新数据在内的 N 个数据的算术平均值或加权平均值，就可得到新的滤波值。这样，每进行一次采样，就可以计算出一个新的平均值，也即测量数据取一丢一，测量一次便计算一次平均值，大大地加快了数据处理的能力。这种数据存放方式可以用环形队列结构方便地实现，即每存入一个新的数据便自动冲去一个最老的数据。

滑动平均滤波算法的最大优点是实时性比较好，同时提高了系统的响应速度。

4）去极值平均滤波

算术平均滤波对抑制随机干扰效果好，但不能将明显的偶然的脉冲干扰消除，只是把其采集进而平均到采样结果中，因此明显的脉冲干扰会使平均值远离实际值，降低了测量精度。去极值平均滤波的算法是：连续采样 N 次，去掉一个最大值，去掉一个最小值，然后再计算余下的 $N-2$ 个采样数据的算术平均值。

在编制算法程序时，为了便于用移位来代替除法求得平均值，$N-2$ 应取 2、4、8 等，因此 N 应取 4、6、10 等。

4. 惯性滤波

前面介绍的基本上属于静态滤波，主要适用于变化比较快的信号参数，如压力、流量、速度等。对于慢速随机变化的信号，则采用在短时间内连续采样求平均值滤波的方法，其滤波效果往往并不理想。为了提高滤波效果，通常可采用动态滤波方法，也即惯性滤波，通常也称为一阶滞后滤波方法。惯性滤波是模拟硬件 RC 低通滤波器的数字实现，可以将普通硬件 RC 低通滤波器的微分方程用差分方程来表示，用软件算法来模拟硬件滤波器的功能。

典型的一阶低通 RC 模拟滤波器的结构如图 8.9 所示。由图 8.9 可以写出一阶低通模拟滤波器的传递函数，即

R
X(s)
C
Y(s)

图 8.9　RC 低通滤波器

$$G(s)=\frac{Y(s)}{X(s)}=\frac{1}{T_f s+1} \tag{8.5}$$

式(8.5)中，$T_f=RC$ 是滤波器的滤波时间常数，其大小直接关系到滤波效果。一般来说，T_f 越大，则滤波器的截止频率也即滤出的干扰频率越低，但由于大时间常数和高精度的 RC 电路不易制作与实现，因为时间常数 T_f 愈大，必然要求电容 C 也愈大，其漏电流也随之增大，从而使 RC 网络的误差增大和设备体积增大，所以硬件 RC 滤波器不可能对极低频率的信号进行滤波。但用软件做成的低通数字滤波器，就能很好地克服上述缺点，在滤波常数要求大的场合，这种方法尤其实用。

将式(8.5)写成差分方程形式

$$T_f\frac{Y(k)-Y(k-1)}{T}+Y(k)=X(k) \tag{8.6}$$

式(8.6)整理后可得

$$Y(k)=\frac{T_f}{T_f+T}Y(k-1)+\frac{T}{T_f+T}X(k)=(1-\alpha)Y(k-1)+\alpha X(k) \tag{8.7}$$

所以可得惯性滤波的表达式为

$$Y(k)=(1-\alpha)Y(k-1)+\alpha X(k) \tag{8.8}$$

式中，$X(k)$为第 k 次滤波输入值，也即第 k 次采样值；$Y(k)$为第 k 次采样的滤波输出值；$Y(k-1)$为第 $k-1$ 次采样的滤波输出值；α 为滤波平滑系数，$\alpha=T/(T+T_f)$；T 为采样周期；T_f为滤波环节的时间常数。

通常采样周期远小于滤波环节的时间常数，也就是输入信号的频率快，而滤波环节时间常数相对较大，这也是一般滤波器的概念，所以这种滤波方法相当于 RC 滤波器。

T 和 T_f的选择可根据具体情况来确定，滤波系数可根据用户选定的 T 和 T_f计算得到，一般 α 远小于 1。同时根据一阶惯性滤波的频率特性，若滤波系数越大，则带宽越窄，滤波

频率也越低。因此，应根据实际情况，选择适当的 α 数值，使得被滤波的信号既不产生明显的纹波，延时又不太大。

5. 复合滤波

为了进一步提高滤波效果，常常可采用复合滤波程序，也就是把两种以上的滤波方法结合起来使用，比如把中值滤波的思想与算术平均滤波两种结合起来，就可得到一种常用的复合滤波法。其具体实现方法是首先将 N 个采样值按大小排序，然后分别去掉最大值和最小值，再把剩下的值加起来，最后取平均值。

显然，这种滤波算法兼有算术平均滤波和中值滤波的优点。它既可以去掉脉冲干扰，又可以对采样值进行平滑处理，因而在快速和慢速参数变化的系统中均能适用。

6. 各种滤波方法的比较

以上主要讨论了几种数字滤波方法的滤波原理，每种滤波程序都各有其特点，在实际应用中，虽然数字滤波是消除计算机控制系统干扰的好方法，但一定要注意，并不是任何一个系统中都需要进行数字滤波。究竟采用不采用，以及到底采用哪一种数字滤波方法，应视具体情况而定。可能有的系统并不需要进行数字滤波，或者应用得不恰当，非但不能达到滤波的效果，而且还会降低控制品质，造成不良影响；有的系统可能采用复合滤波的方法将会取得更好的滤波效果。在实际选用时可从以下两个方面来考虑。

1）滤波效果

一般来说，对于变化比较缓慢的参数比如温度，可选用程序判断滤波和一阶滞后滤波；对于变化比较快的信号比如压力、流量等，则可选用算术平均滤波和加权平均滤波，特别是加权平均滤波比算术平均滤波更好；对于要求比较高的系统则可选用复合滤波方法。

2）滤波时间

在考虑滤波效果的前提下，应尽量采用执行时间比较短的滤波程序，如果计算机的时间允许，则可采用更好的复合滤波程序。

总之，随着计算机控制技术的发展，数字滤波方法将越来越完善，并且还会不断地有新的滤波方法出现，各种数字滤波算法各有优缺点，在实际应用中应根据具体情况将其有机结合起来，为数据处理选择一种最优最合适的滤波算法，保证数据准确、快速地反映被检测对象的实际情况，为生产管理提供有效的数据。

8.4.3 非线性处理技术

在数据处理系统中，一般总希望系统的输出和输入呈简单的线性关系，特别当用仪表来检测和显示某个物理量时，如果呈线性关系就能得到均匀的刻度，这样不仅读起数来清楚方便，而且使仪表在整个刻度范围内灵敏度一致，从而便于读数以及对系统进行处理。但在实际工程应用中，计算机从模拟量输入通道得到的有关现场信号与该信号所代表的物理量之间不一定成线性关系，经常存在着非线性关系。比如热电阻或热电偶与温度的关系就是非线性的；在流量测量中，流经孔板的差压信号与流量之间成平方根关系，也是非线性的。为了保证这些参数能够按线性输出，必须进行非线性补偿，将输出信号与被测物理量之间的非线性关系补偿为线性关系，这也称为线性化处理，这样可以提供变换精度，便于对系统进行

分析和处理。

在常规的自动化仪表中，常引入线性化器来补偿其他环节的非线性，如凸轮机构及曲线板、非线性电位计、二极管阵列、运算放大器等，这些方法均属于硬件补偿，不但成本高，而且使设备更加复杂，精度一般也不太高，并且对有些误差的补偿是极为困难的，有些甚至是不可能的。而在计算机控制系统中，用计算机进行非线性补偿，即用软件代替硬件实现非线性补偿，不仅能节省大量的硬件开支，而且补偿方法灵活方便，精度也大为提高，因而得到了广泛的应用。软件实现非线性补偿的方法主要有计算法、查表法、插值法与折线法。

1. 计算法

在工程实际中，当参数间的非线性关系可以用数学方程来表示时，计算机可以按公式进行计算，实现对非线性补偿。在过程控制中较为常见的两个非线性关系是差压与流量、温度与热电势。

1) 孔板差压与流量的关系

用孔板测量气体或液体流量时，差压变送器输出的孔板差压信号 ΔP，与实际流量 F 之间呈平方根关系，即

$$F = k\sqrt{\Delta P} \tag{8.9}$$

式中，k 是流量系数，它和孔板及被测流体的温度、压力有关。计算机必须把差压信号 ΔP 平方后，才能和流量成线性关系。为了计算平方根，实用中可采用牛顿迭代法来计算平方根。设

$$y = \sqrt{x} \quad (x > 0)$$

则

$$y(k) = \frac{1}{2}\left[y(k-1) + \frac{x}{y(k-1)}\right] \tag{8.10}$$

式中，$y(k)$是第 k 次迭代所得的平方根；$y(k-1)$是第 $k-1$ 次迭代所得的平方根；x 为被开方数。关于牛顿迭代法的原理以及初始值的选取读者可以参考有关文献。

2) 热电偶的热电势同所测温度的关系

在温度测量中大量使用热电偶作为测温元件，但热电偶的热电势 E 同所测温度 T 之间也是呈现非线性关系，而且不同种类的热电偶的非线性程度还不一样。但一般情况下，常见的 E—T 关系都可用高次多项式表示为

$$T = a_4E^4 + a_3E^3 + a_2E^2 + a_1E + a_0 \tag{8.11}$$

式中，T 为温度，E 为热电偶测量的热电势；a_0、a_1、a_2、a_3、a_4 为热电偶系数，不同种类的热电偶以及不同的测温范围，其取值是不一样的，运算时常作为常数处理。将式(8.11)略加变换处理可得

$$T = \{[(a_4E + a_3)E + a_2]E + a_1\}E + a_0 \tag{8.12}$$

式(8.12)可将原来非线性化的关系分成多个线性化的式子来实现，并且可以方便地用程序来实现。

2. 查表法

在计算机控制系统中，有些参数的计算相对来说非常复杂，如果用计算法计算不仅实现

的程序长，难于计算，而且还会耗费大量的时间，有时还会涉及到指数、对数、积分以及微分运算，这些算法用程序来编制和实现比较复杂，对于这些非线性关系的解决，可以采用查表法。

所谓查表法，就是事先将计算或测得的数据按一定的顺序编制成表格，存入内存单元中，查表程序的任务根据测量值或中间结果，查出最终所需要的结果。查表是一种非数值的计算方法，利用这种方法可以完成数据的补偿、计算和转换等各种工作。计算机在线的工作量仅仅是根据采样值来查表获得相应的结果，它具有程序简单、执行速度快的优点。

查表程序的繁简程度以及查询时间的长短，除了与表格的长度有关外，表格的排列方法对其影响也很大。一般来说表格有两种排列方法，即数据任意排列的无序表格和数据按一定顺序排列的有序表格。表格的排列不同其查表的方法也不尽相同，但常用的查表方法有计算查表法、顺序查表法以及对分查表法。

(1) 计算查表法要求表格中存放的各元素要严格地按照某种顺序排列，并且要求各元素在表中的排列格式以及所占用的空间必须一致。这种查表法的关键在于找出一个计算表格地址的公式，只要公式存在，查表的时间就与表格的长度无关。只要根据所给的数据元素 X，通过一定的公式计算，求出元素 X 所对应的数值的地址，然后将该地址单元中的内容取出即可，比如常用的数码显示程序、功能键地址转移程序等就是利用计算查表法来实现的。

(2) 顺序查表法适用于无序排列表格的查找，也即按照表格中数据存放的顺序依次进行查找。其查表方法类似于人工查表，因为无序表格中所有项的排列均无一定的规律，所以只能按照顺序从第一项开始逐项寻找，直到找到所要查找的关键字为止。顺序查表法的逐项查找虽然比较“笨”，但对于无序表格或者比较短的表格，仍然是一种比较好的常用的方法。

(3) 前面介绍的两种查表方法中，计算查表法速度比较快，但对于表格的要求比较苛刻，其应用具有一定的局限性；而顺序查表法的速度比较慢。在实际应用中，很多的表格都比较长，如果用前面的查表方法耗时且难以查找，但这些表格都是按照一定的顺序排列，比如热电偶的热电势与温度对照表以及流量测量中的差压与流量对照表等，对于这样的表格，可以采用对分查表法。

对分查表法的具体做法就是每次取表的中间值进行查找排查，然后在剩余的一半中再进行对分查找，也即先取表格的中间值 $D=N/2$ 进行查找，与要查找的 X 值进行比较，如果相等则查到，假设表格是按从小到大的顺序排列，如果 $X>N/2$，则下一次取 $N/2\sim N$ 的中值，也即 $3N/4$ 再进行比较；如果 $X<N/2$，则下一次取 $0\sim N/2$ 的中值，也即 $N/4$ 再进行比较；如此比较循环下去，就可以逐次逼近要查找的元素，直到找到为止。

3. 插值法

计算机非线性处理应用最多的方法就是插值法。插值法的实质是找出一种简单的、便于计算处理的近似表达式代替非线性参数，用这种方法得到的公式叫做插值公式。常用的插值公式有线性插值公式、拉格朗日插值公式、多项式插值公式等。但由于计算机控制系统对系统的实时性要求较高，在利用插值逼近方法进行线性化处理的应用中使用最多的是运算量较小的线性插值法，其计算简单、快捷，如果精度要求较高并且计算时间允许的话，可以完全采用二次曲线插值法来拟合曲线段。下面重点介绍多项式插值计算方法。

插值多项式是用一个 n 次多项式来代替某种非线性函数关系的方法，其插值原理是：被测参数 y 与传感器的输出值 x 具有 $y=f(x)$ 的函数关系，假设知道在 $n+1$ 个相异点处的函数值分别为 $f(x_0)=y_0, f(x_1)=y_1, \cdots, f(x_n)=y_n$，现构造一个 n 次多项式，即

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \tag{8.13}$$

使其最大限度地逼近 $f(x)$，并且在 $x_i(i=0\sim n)$ 处与 $f(x_i)$ 相等，则函数 $P_n(x)$ 被称为 $f(x)$ 的插值函数，x_i 被称为插值点。把 $y=f(x)$ 中这 $n+1$ 个相异点处的值作为插值代入 n 次多项式 $P_n(x)$，便可以得到 $n+1$ 个一次方程组

$$\begin{cases} a_n x_0^n + a_{n-1} x_0^{n-1} + \cdots + a_1 x_0 + a_0 = y_0 \\ a_n x_1^n + a_{n-1} x_1^{n-1} + \cdots + a_1 x_1 + a_0 = y_1 \\ \vdots \\ a_n x_n^n + a_{n-1} x_n^{n-1} + \cdots + a_1 x_n + a_0 = y_n \end{cases} \tag{8.14}$$

式(8.14)中，x_0、x_1、…、x_n 为已知传感器的被测值；y_0、y_1、…、y_n 为已知的被测参数；通过方程组可以求解系数 a_0、a_1、…、a_n，将其代入式(8.12)就可以构造成一个可代替这种函数关系的可插值的多项式 $P_n(x)$，式(8.13)也称为函数 $f(x)$ 以 x_0、x_1、…、x_n 为基点的插值多项式。

4. 折线法

一般来说，计算法和插值法都可能会带来大量的运算，特别对于小型工业控制机来说，占用的内存也比较大。为了简单起见，可以进行分段线性化，也即用多段折线代替曲线。

分段线性化的过程是：首先判断测量数据处于哪一折线段内，然后按相应段的线性化公式计算出线性值。折线段的分法并不是唯一的，可以视具体情况和要求来确定。当然，折线段数越多，线性化精度越高，但是软件的开销也相应增加。

8.4.4 标度变换技术

在计算机控制系统中，生产过程中的各个参数都有不同的数值和量纲，例如温度的单位为℃，压力的单位为 Pa，流量的单位为 m^3/h，电压的单位为 V，这些参数都经过变送器转换成 A/D 转换器能够接收的电压信号，又由 A/D 转换成 00～FFH(8 位)的数字量，它们不再是带量纲的参数值，而是仅代表参数值的相对大小，这些数字量并不一定等于原来带有物理量纲的参数值。比如温度的变换范围是 0～100℃，如果转换成数字量则 FFH 表示的是 100℃，而如果变换范围是－200～200℃时，则数字量 FFH 表示的是 200℃。

为了方便操作人员管理操作以及满足一些运算、显示、记录和打印的要求，必须把这些数字量转换成相应的带有量纲的物理量数值，这种转换就是所谓的标度变换或工程量转换。标度变换的任务就是把计算机系统检测的对象参数的二进制数值还原变换为原物理量的工程实际值。图 8.10 为标度变换原理图，这是一个温度测量系统，采用某种热电偶传感器把现场温度 0～1000℃转变为 0～48mV 的电压信号，再经过输入通道中的运算放大器放大到 0～5V，再经由 8 位 A/D 转换成 00～FFH 的数字量，这一系列的转换过程是由输入通道的硬件电路来完成的。CPU 在将读入的数字信号送到显示器进行显示处理以前，必须把这一无量纲的二进制数值再还原变换成原来量纲为℃的温度信号。比如最小值 00H 应变换对

应 0℃,最大值 FFH 应变换对应 1000℃。

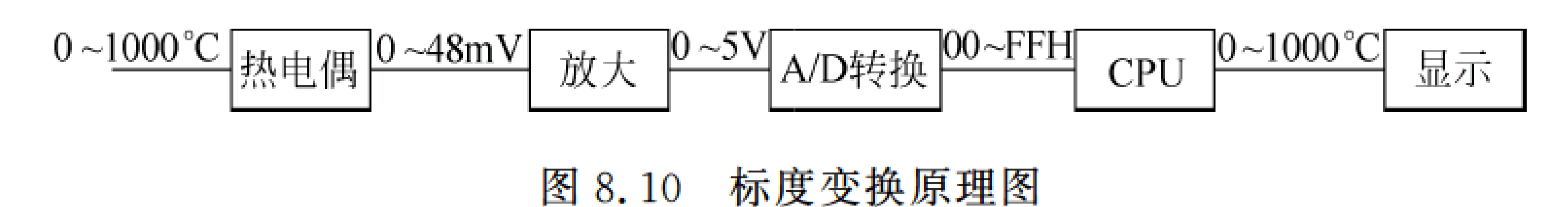


图 8.10　标度变换原理图

标度变换的过程是由算法软件程序来实现的。标度变换有各种类型,它取决于被测参数和测量传感器或变送器的类型,应根据实际情况选用适当的标度变换方法进行设计。

1. 线性标度变换

线性标度变换是最常用的标度变换方式,这种变换的前提条件是被测参数与 A/D 转换结果之间为线性关系,如图 8.11 所示,其标度变换公式为:

$$A_x = A_0 + (A_m - A_0)\frac{N_x - N_0}{N_m - N_0} \tag{8.15}$$

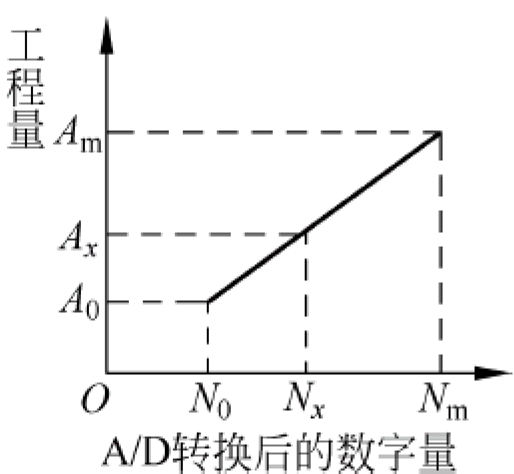


图 8.11　线性关系标度变换示意图

式中,A_0为一次测量仪表的下限,也即测量范围的最小值;A_m为一次测量仪表的上限,也即测量范围的最大值;A_x为实际测量值(工程量);N_0为仪表下限对应的数字量;N_m为仪表上限对应的数字量;N_x为测量值所对应的数字量。

式(8.15)为线性标度变换的通用公式,其中 A_0、A_m、N_0、N_m 对于某一个固定的被测参数来说是常数,不同的参数有不同的值。为使程序设计简单,一般把被测参数的起点 A_0(输入信号为 0)所对应的 A/D 输出值设为 0,即 $N_0=0$,这样式(8.15)可简化为

$$A_x = \frac{N_x}{N_m}(A_m - A_0) + A_0 \tag{8.16}$$

在很多测量系统中,仪表的测量下限值 $A_0=0$,此时,其对应的输出数字量 $N_0=0$,则式(8.16)可进一步的简化为

$$A_x = A_m\frac{N_x}{N_m} \tag{8.17}$$

有时,工程量的实际值还需经过一次变换,如电压测量值是电压互感器的二次测量的电压,则与其一次测量的电压还有一个互感器的变比问题,这时上式应再乘上一个比例系数 k,即

$$A_x = k \cdot \left[\frac{N_x}{N_m}(A_m - A_0) + A_0\right] \tag{8.18}$$

在计算机控制系统中,为了实现上述转换,可把它们设计成专门的子程序,把各个不同的参数所对应的 A_0、A_m、N_0、N_m 存放在存储器中,然后当某一参数需要进行标度变换时,只需调用标度变换子程序即可。

【例 8.2】 已知某热处理炉温度测量仪表的量程为 0～800℃,在某一时刻计算机经检测、A/D 采样变换和数字滤波后得到的数字量为 CDH。假设该仪表的量程是线性的,A/D 转换器是 8 位的,求此时的温度值是多少?

解:由于 $A_0=0$℃,$A_m=800$℃,$N_x=\text{CDH}=(205)_D$,$N_m=\text{FFH}=(255)_D$,所以根据式(8.16)可知此时的温度为

$$A_x = \frac{N_x}{N_m}(A_m - A_0) + A_0 = \frac{205}{255}(800 - 0) + 0 = 643.13℃$$

2. 非线性参数标度变换

必须指出，上面介绍的标度变换程序，它只适用于具有线性刻度的参量。如果传感器的输出信号与被测参数之间呈非线性关系时，就需要建立新的标度变换公式。由于非线性参数的变化规律各不相同，所以其标度变换公式应根据具体问题具体分析，首先求出它所对应的标度变换公式，然后再进行设计。

在过程控制系统中，最常见的非线性关系是用孔板测量气体或液体的流量时，其差压变送器输出的孔板差压信号 ΔP，同实际流量 F 之间呈平方根关系，即

$$F = k\sqrt{\Delta P} \tag{8.19}$$

式中，k 是流量系数，它和孔板及被测流体的温度、压力有关。

由式(8.19)可知，流体的流量与被测流体流过节流装置时前后产生的压力差的平方根成正比，于是可得到测量时的标度变换公式

$$\frac{F_x - F_0}{F_m - F_0} = \frac{k\sqrt{N_x} - k\sqrt{N_0}}{k\sqrt{N_m} - k\sqrt{N_0}}$$

也即

$$F_x = \frac{\sqrt{N_x} - \sqrt{N_0}}{\sqrt{N_m} - \sqrt{N_0}}(F_m - F_0) + F_0 \tag{8.20}$$

式中，F_0为流量仪表下限值；F_m为流量仪表上限值；F_x为被测量的流量值；N_0为差压变送器下限所对应的数字量；N_m为差压变送器上限所对应的数字量；N_x为差压变送器所测得的差压值(数字量)。

式(8.20)为流量测量中标度变换的通用表达式，对于流量仪表，一般下限 $F_0=0$，则 $N_0=0$，这样式(8.20)可以简化为

$$F_x = \frac{\sqrt{N_x}}{\sqrt{N_m}}F_m \tag{8.21}$$

按式(8.21)即可设计一个流量测量值与工程值的转换程序。

许多非线性传感器并不像差压变送器那样，可以写成一个简单的公式，或者虽然能够写成公式，但是计算相当困难。这时可采用多项式插值，建立一个比较简单的表达式，然后再进行标度变换。

8.4.5 越限报警处理

为了实现安全生产，在计算机控制系统中，对于重要的参数和部位一般都设有上下限检查和紧急状态报警系统，以便及时提醒操作人员注意，必要时采取相应的紧急措施，以保证生产过程的正常运行和操作者的人身安全。在计算机控制系统中多采用声、光以及语音来进行报警，必要时可以增加打印记录功能，以便记录报警的参数、发生的时间和事故的情况等。

报警处理的设计方法有两种，一种是采用硬件申请、软件处理的报警程序，这种方法的报警极限参数是直接由相应的传感器来产生的，例如温控开关式报警装置，当温度高于或低

于某一温度值时，开关接点即闭合（假如正常时断开），可利用这些开关信号，向控制 CPU 申请中断，实现系统报警的功能。另一种是全软件法，这种方法是把计算机采样读入的数据进行相应的预处理、数字滤波和标度变换等之后，再与给定的上下限进行比较，根据比较结果，如果高于上限值或低于下限值，则进行报警处理，否则就作为采样的正常值进行显示和控制。上下限的规定值可以作为常数存于内存中。

越限报警是工业控制系统中一种常见而又实用的报警形式，它将采样得到的数据经计算机做一定的处理后与规定的工艺参数范围的上下限约束值进行比较，根据生产工艺的要求，可分为上限报警、下限报警和上下限报警。在控制系统中，报警参数可以是输入偏差 e、控制量 u、被控参数 y 或被测参数等，假设需要报警的参数是 x，该参数的上限、下限约束值分别为 x_{max} 和 x_{min}，则越限报警有如下几种形式。

(1) 上限报警。若 $x > x_{max}$，则发出上限报警，否则继续执行原定操作。

(2) 下限报警。若 $x < x_{min}$，则发出下限报警，否则继续执行原定操作。

(3) 上下限报警。若 $x > x_{max}$，则上限报警，否则判断 $x < x_{min}$ 否？若是则下限报警，否则继续执行原定操作。

同时在具体设计报警程序时，如果测量值在极限值附近波动时则会发生频繁的报警，造成操作人员人为地感到紧张。实际上检测参数的上、下限的值都不是唯一的，大都有一个称之为回差带的范围。为了避免测量值在报警极限点处来回摆动造成频繁报警，一般应在极限值附近设置一个回差带 2ε，如图 8.12 所示。只有当被测值超越 $x_{max}+\varepsilon$ 时，才算越过上限报警值并输出报警信号；当被测值下降到 $x_{max}-\varepsilon$ 时，则复位并撤销上限报警信号。同理，当被测值低于 $x_{min}-\varepsilon$ 时，才算越过下限并设置输出相应的报警信号；当被测值上升到 $x_{min}+\varepsilon$，才撤销越下限报警信号。这样，回差值 2ε 避免了测量值在极限值 x_{max} 和 x_{min} 一点处来回摆动造成频繁报警，ε 值的大小可根据现场具体的被测参数设定。

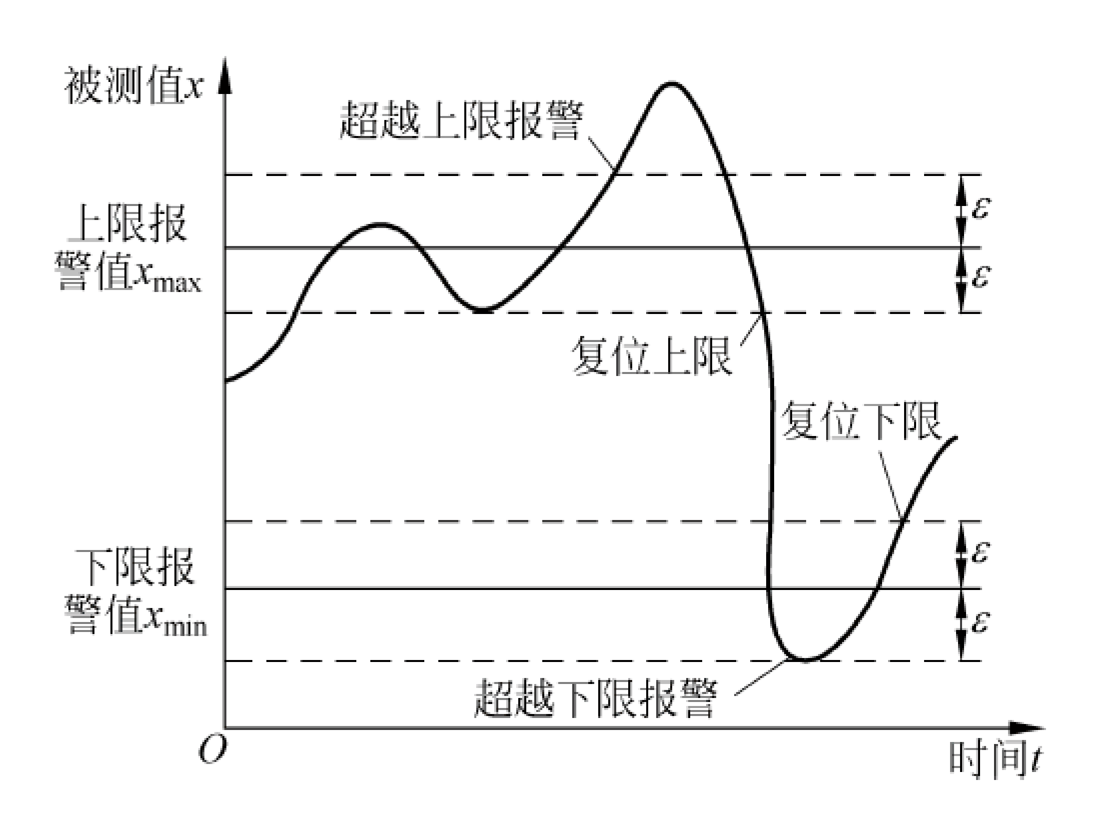


图 8.12 越限报警范围示意图

习题

1. 计算机控制系统中对软件的要求有哪些？
2. 计算机控制系统的软件功能有哪些？
3. 什么是模块化程序设计和结构化程序设计？

4. 简要说明应用软件的设计流程及方法。

5. 简要说明计算机控制系统中控制策略软件的功能。

6. 计算机控制系统中常用的显示方式有哪些?

7. 简述8段LED显示器段选码的概念以及0~F段选码的确定。

8. 分别说明LED静态显示和动态显示的原理。

9. 什么是OPC技术?

10. 何为数字调零和系统校准?它们是怎样实现的?

11. 什么是数字滤波?数字滤波与模拟 *RC* 滤波相比有什么特点?

12. 常用的数字滤波方法有哪些?分别适用于什么场合?

13. 在计算机控制系统中,非线性补偿的方法有哪些?简要说明其实现的原理。

14. 什么是标度变换?试写出线性标度变换的计算公式。

15. 已知某热处理炉温度的变化范围为0~1000℃,经过温度变送器后的输出电压为1~5V,再经过ADC0809转换和数字滤波后得到的数字量为9CH。其中ADC0809的输入范围为0~5V,假设该仪表的量程是线性的,求此时所对应的温度值是多少?

16. 在计算机控制系统中,为什么要设置越限报警?越限报警有哪些方式?

第9章 组态软件应用

随着工业自动化水平的迅速提高,工业通用自动化组态软件能使用户根据自己的控制对象和控制目的任意地组态,很方便地完成自动化控制工程。本章简单介绍工业组态软件的概况,并以国内著名的组态软件——组态王软件为例,重点介绍组态王的基本功能使用以及控件和报表的应用,详细讲解基于组态王软件的监控系统应用实例。

9.1 工业组态软件概述

9.1.1 概述

随着工业自动化技术和计算机的发展,越来越需要计算机对现场控制设备(比如 PLC、智能仪表、板卡、变频器等)的监控功能,于是数据采集与监视控制(Supervisory Control And Data Acquisition,SCADA)系统应运而生。凡是具有数据采集和系统监控功能的软件,都可以称为 SCADA 软件,它是建立在 PC 基础之上的自动化监控系统,SCADA 系统的应用领域很广,它可以应用于电力系统、航空航天、石油、化工等领域的数据采集与监视控制以及过程控制等诸多领域。

工业组态软件通常简称为组态软件,它是可以快速地建立计算机监控系统界面的软件工具,组态软件通常运行于工业控制计算机平台,同时与各类控制设备一起组成计算机监控系统。

组态(Configuration)的含义是设置、配置,是指操作人员根据用户需求及控制任务的要求,使用软件工具,对计算机资源进行组合以达到应用的目的。组态过程可以看作是配置用户应用软件的过程,软件提供了各种"零部件"模块供用户选择,采用非编程的"搭积木"操作方式,主要通过参数填写、图形连接和文件生成等,组合各功能模块,构成用户应用软件。控制工程师可以在不必了解计算机的硬件和程序的情况下,把主要精力放在控制对象和算法上,而不是形形色色的通信协议和复杂的图形处理上。

组态软件产生于 20 世纪 80 年代,世界上第一个商品化监控组态软件是美国 Wonderware 公司开发的 InTouch 软件,也是最早进入我国的组态软件。在 20 世纪 90 年代中后期以来,组态软件在我国逐渐得到了广泛的应用普及,随着组态软件技术的快速发展,实时数据库、实时控制、SCADA、通信及联网、开放数据接口、多 I/O 设备的广泛支持已经成为它的主要内容,在工业自动化领域将会得到越来越广泛的应用。

9.1.2 组态软件的基本特点

通用组态软件具有实时多任务、接口开放、使用灵活、运行可靠的特点，下面就其主要特点做简单介绍。

1. 实时多任务

组态软件可以在一台计算机上同时完成数据采集与输出、数据处理与算法实现、图形显示及人机对话、实时数据的存储、检索管理、实时通信等多个任务，处理功能强大。

2. 封装性

通用组态软件所能完成的主要功能都用一种方便用户使用的方法封装起来，对于用户，不需了解太多的编程语言技术，就能又快又好地完成一个复杂工程所要求的几乎所有功能，使用简单方便。

3. 延续性和可扩展性

用通用组态软件开发的应用程序，当现场设备或用户需求发生改变时，不需要从底层作很多修改就可以方便地完成软件的更新和升级，为后续的使用带来极大的便捷。

4. 通用性

使用者可以根据工程实际情况，利用通用组态软件提供的丰富的底层设备驱动、开放式的数据库和强大的界面制作工具，就能完成一个具有实时数据处理、动画显示效果、历史数据和曲线并存等功能的工程，不受所属行业限制，并且底层设备的驱动也在不断地更新完善中，适应新的技术的发展。

9.1.3 典型组态软件介绍

组态软件自 20 世纪 80 年代初期诞生至今已经有三十多年的发展历程。早期的组态软件大都运行在 DOS 环境下，其特点是具有简单的人机界面、图库和绘图工具箱等基本功能，图形界面的可视化功能不是很强大。随着微软 Windows 操作系统的发展和普及，Windows 下的组态软件成为主流。

目前，世界上有不少专业厂商生产和提供各种组态软件产品，市面上的软件产品种类繁多，各有所长，应根据实际工程需要加以选择。组态软件国产化的产品近年来比较出名的有组态王、世纪星、力控、MCGS、易控等，国外主要产品有美国 Wonderware 公司的 InTouch、美国 GE Fanuc 智能设备公司的 IFix、德国西门子公司的 WinCC 等。下面简单介绍几种典型的组态软件。

1. 组态王

组态王 Kingview 软件是国内具有自主知识产权、市场占有率高、影响比较大的组态软件。该组态软件提供了资源管理器式的操作主界面，使用方便，操作灵活。组态王软件还提

供了多种硬件驱动程序,支持众多的硬件设备。应用领域几乎囊括了大多数行业的工业控制,已广泛应用于化工、电力、邮电通信、环保、水处理、冶金和食品等行业。

2. 力控

北京三维力控科技有限公司的 ForceControl(力控)组态软件也是国内出现较早的组态软件之一,具有一定的市场占有率。公司产品主要有力控通用版和电力版组态软件,适应于不同领域的应用,并且功能丰富,实用性和易用性都比较好。

3. WinCC

WinCC(Windows Control Center,视窗控制中心)是德国西门子公司开发的一套完备的组态开发环境。WinCC 监控系统可以运行在 Windows 操作系统下,使用方便,具有生动友好的用户界面,还能链接到别的 Windows 应用程序(如 Microsoft Excel 等)。WinCC 是一个开放的集成系统,既可独立使用,也可集成到复杂、广泛的自动控制系统中使用。同时内嵌 OPC 技术,可对分布式系统进行组态。

4. InTouch

美国 Wonderware 的 InTouch 软件是最早进入我国的组态软件。在 80 年代末 90 年代初,基于 Windows 3.1 的 InTouch 软件曾让我们耳目一新,最新的 InTouch 7.0 版已经完全基于 32 位的 Windows 平台,并且提供了 OPC 支持。InTouch 软件的图形功能比较丰富,使用比较方便,其 I/O 硬件驱动丰富,工作稳定,在国内市场也普遍受到欢迎。

9.2　组态王基本功能的实现

9.2.1　组态王软件基本组成

组态王软件由工程管理器、工程浏览器、画面运行系统和信息窗口 4 部分组成。其中,工程管理器实现对组态王各种版本工程的集中管理,使用户在进行工程开发、工程备份和数据词典的管理上方便了许多,其主要作用就是为用户集中管理本机上的所有组态王工程。

工程浏览器是组态王的一个重要组成部分,它将图形画面、命令语言、设备驱动程序、配方、报警、网络等工程元素集中管理,工程人员可以一目了然地查看工程的各个组成部分。组态王工程浏览器内嵌画面开发系统,也即组态王开发系统,是应用程序的集成开发环境,工程人员在这个环境里进行系统开发。工程浏览器和画面运行系统是各自独立的 Windows 应用程序,均可单独使用,两者又相互依存,在工程浏览器的画面开发系统中设计开发的画面应用程序必须在画面运行系统环境中才能运行。

组态王信息窗口是一个独立的 Windows 应用程序,用来记录、显示组态王开发和运行系统在运行状态时的信息。信息窗口中显示的信息可以作为一个文件存于指定的目录中或是用打印机打印出来,供用户查阅。当工程浏览器、画面运行系统等启动时,会自动启动信息窗口。

9.2.2 组态王应用程序建立过程

建立组态王应用程序工程的一般过程如下。

(1) 设计图形界面(定义画面)。

(2) 定义设备。

(3) 构造数据库(定义变量)。

(4) 建立动画连接。

(5) 运行和调试。

需要说明的是,这 5 个步骤并不是完全独立的,事实上,这 5 个部分常常是交错进行的。在用组态王画面开发系统构造开发工程时,主要考虑以下 3 个方面。

(1) 图形制作。用户希望得到怎样的图形画面?也就是怎样用抽象的图形画面来模拟实际的工业现场和相应的工控设备,从而使操作灵活、方便直观。用组态王系统开发的应用程序是以“画面”为基本单位,每一个画面对应于程序实际运行时的一个 Windows 窗口。

(2) 数据定义。怎样用数据来描述工控对象的各种属性?也就是创建一个具体的实时数据库,此数据库中的变量反映了工控对象的各种属性,比如对象的温度、压力等,还包括操作者指令的变量,比如阀门开度、电源开关、指示灯亮灭等。变量可以是模拟的,也可以是离散的,可以是内存型的,也可以是 I/O 型的。工程人员需要在设计中认真地规划工程中的数据,进行正确的定义,可能还要为临时变量预留空间。

(3) 动画连接。数据和图形画面中的图素的连接关系是什么?也就是画面上的图素以怎样的动画来模拟现场设备的运行,以及怎样让操作者输入控制设备的指令。

9.2.3 建立一个新项目

在组态王中,用户开发的每一个应用系统称为一个项目,每个项目必须存放在一个独立的目录中,不同的项目不能共用一个目录。项目存放的目录称为工程路径,在每个目录下,除了用户制作的一些画面文件外,组态王还为此项目生成了一些重要的数据文件,这些数据文件一般不允许修改。

在建立一个新的应用程序时,都必须先为这个应用程序指定存放的目录,以便组态王根据工程路径对不同的应用程序进行不同的自动管理。下面建立一个计数器监控系统,其实现的目标是:计数器从零开始每隔 2s 加 1,并把当前的计数值显示在当前监控画面上,如果计数器数值达到设置的最大值 100 时,监控画面中对应的状态指示灯由绿色变为红色,并可以通过“清空”按钮将计数器的值清空又重新开始计数;单击监控画面中的“关闭”按钮时,结束系统的程序运行监控。建立这个新项目的过程如下。

(1) 启动组态王工程管理器,单击菜单栏“文件\新建工程”命令或单击快捷工具栏“新建”命令后,弹出“新建工程向导之一——欢迎使用本向导”对话框,如图 9.1 所示。

(2) 单击“取消”按钮退出新建工程向导。单击“下一步”按钮继续新建工程。弹出“新建工程向导之二——选择工程所在路径”对话框,如图 9.2 所示。在该对话框中,可以输入新建工程的路径,如果输入的路径不存在,系统将自动提示用户。如果要更改工程路径,可以单击“浏览”按钮,从弹出的路径选择对话框中选择合适的工程存放路径。



图 9.1　新建工程向导一



图 9.2　新建工程向导二

(3) 单击“上一步”按钮返回上一页向导对话框。单击“取消”按钮退出新建工程向导。单击“下一步”按钮进入新建工程向导三，弹出“新建工程向导之三——工程名称和描述”对话框，如图 9.3 所示。在“工程名称”文本框中输入新建工程的名称，名称有效长度小于 32 个字符，这个是必须输入的，比如在对话框的工程名称中输入“计数器监控系统”。在“工



图 9.3　新建工程向导三

程描述”中输入对新建工程的描述文本，描述文本有效长度小于40个字符，这个是可选的，比如可以输入“监控当前计数器的值”。并且注意在组态王中，工程名称是唯一的，不能重名，工程名称和工程路径也是一一对应的。

(4) 单击“上一步”按钮返回向导的上一页。单击“取消”按钮退出新建工程向导。单击“完成”按钮确认新建的工程，弹出“是否将新建的工程设为组态王当前工程”对话框，单击“是”确认将新建的工程设置为组态王当前工程，单击“否”不改变当前工程的设置。此时组态王工程管理器中出现新建的工程，如图9.4所示。其新建工程的路径是在向导二中指定的路径，在该路径下会以工程名称为目录建立一个文件夹。至此完成新建工程操作。



图9.4 “工程管理器”界面

完成以上4个步骤的操作就可以新建一个组态王工程的工程信息了。此处在实际上并未真正创建工程，只是在用户设定的工程路径下设置了工程信息，只有用户将此工程作为当前工程，且切换到组态王开发环境时才是真正创建工程。

9.2.4 开发环境——工程浏览器

工程浏览器是组态王软件的核心部分和管理开发系统，在工程浏览器中可以查看工程的各个组成部分，并且可以方便地完成项目数据库的构造和外部设备的定义连接等。

在图9.4所示的工程管理器界面中，双击“计数器监控系统”工程，或者选中“计数器监控系统”工程再单击快捷工具栏上的“开发”按钮，出现演示方式提示对话框，单击“确定”按钮，进入到工程浏览器界面，如图9.5所示。



图9.5 “工程浏览器”界面

1. 工程浏览器基本结构

工程浏览器由 Tab 页标签、菜单栏、工具栏、工程目录显示区、目录内容显示区和状态栏构成。

工程浏览器左侧是“工程目录显示区”，以树形结构图显示功能节点，用户可以扩展或收缩工程浏览器中所列的功能项，主要作用是展示工程的各个组成部分。主要包括“系统”、“变量”、“站点”和“画面”4 部分，这 4 部分的切换是通过工程浏览器最左侧的 Tab 标签实现的。其中“系统”部分共有文件、数据库、设备、系统配置、SQL 访问管理器和 Web 等 6 大项。“变量”部分主要进行变量管理，包括变量组。“站点”部分显示定义的远程站点的详细信息，“画面”部分用于对画面进行分组管理，创建和管理画面组。

工程浏览器右侧是“目录内容显示区”，显示每个工程组成部分的详细内容，同时对工程提供必要的编辑修改功能。

工具栏按钮是工程浏览器中菜单命令的快捷方式，每一个按钮对应一个菜单命令。当鼠标放在工具栏的任一按钮上时，立刻会出现提示信息框标明此按钮的功能。

2. 图形画面的制作

画面开发系统是应用程序的集成开发环境，用户在这个环境里进行系统的开发。

1）画面的新建

使用工程管理器新建一个组态王工程后，进入组态王工程浏览器，接下来就是根据工程实际新建组态王画面。新建画面的方法有 3 种。

（1）在“系统”标签页的“画面”选项下新建画面。

单击工程浏览器左边“工程目录显示区”中“画面”项，在右面“目录内容显示区”中显示“新建”图标，鼠标双击该图标，弹出“新画面”对话框，或者右击“新建”画面图标，在弹出的快捷菜单中选择“新建画面”命令，弹出“新画面”对话框，如图 9.6 所示。

（2）在“画面”标签页中新建画面。

右击“新建”画面图标，在弹出的快捷菜单中选择“新建画面”命令，弹出“新画面”对话框，如图 9.6 所示。

（3）单击快捷工具栏“MAKE”按钮或右键单击工程浏览器空白处，从显示的快捷菜单中选择“切换到 Make”命令，进入组态王开发系统。选择“文件\新画面”菜单命令，弹出“新画面”对话框，如图 9.6 所示。

2）画面的制作

在新建画面图 9.6 的属性对话框中，输入画面名称，比如“计数器监控系统”，同时可以设置画面的大小、位置、风格以及画面在磁盘上对应的文件名。该文件名可以由组态王自动生成，也可以根据自己的需要进行修改。设置完成后单击“确定”按钮即进入组态王画面开发系统，此时会自动加载画面制作的工具箱，如图 9.7 所示。

组态王画面开发系统内嵌于组态王工程浏览器中，又称为界面开发系统，是应用程序的集成开发环境，工程人员在这个环境里完成界面设计、动画连接等系统开发工作。组态王画面开发系统的菜单包括文件、编辑、排列、工具、图库、画面以及帮助等，各个主菜单下还有相应的下拉子菜单，分别完成相应的功能，这些菜单的详细使用可以参考组态王帮助文档。



图 9.6 “新画面”对话框



图 9.7 画面开发系统界面

在图 9.7 建立的新画面中可以绘制画面系统所需要的各种图素，绘制图素的主要工具放置在图形编辑工具箱内，如果打开画面时工具箱没有自动出现，可以选择“工具\显示工具箱”或者按 F10 快捷键打开它。工具箱中各基本工具的使用方法和 Windows 中的“画笔”很相似，如果熟悉“画笔”的使用，绘制图形画面比较容易。下面绘制计数器监控系统的简单画面。

(1) 在工具箱中单击文本工具按钮 T，随后将鼠标移动到画面合适的位置上单击即完成了文本框的放置。在文本框中以当前字体可以输入所需文本。输入“计数器监控系统”，输入完毕后单击即完成文字输入。用同样的方法输入需要在文本框中显示的其他内容。

如果要改变文字的字体、字号等，需先选中文本对象，然后在工具箱内选中“字体”图标

或选择菜单“工具\字体”即可出现字体设置对话框。如果需要对输入的文字进行修改，则先选中该文本，右击，在弹出的快捷菜单中单击“字符串替换”，在弹出的“字符串替换”对话框中，输入需要修改的文字即可。若要修改文字的颜色，则可以选中文本，单击“工具箱”中的“显示调色板”按钮，在弹出的“调色板”中单击“字符色”按钮，选择需要的颜色即可进行文字颜色的修改。

(2) 在图形画面中加入计数器状态指示灯对象。可以在画面开发系统中单击 F2 或者选择菜单“图库\打开图库”命令，进入如图 9.8 所示组态王的“图库管理器”界面，单击图库管理器左侧的指示灯库，在右侧对应的指示灯库中选择一个指示灯图形对象，双击所需要的指示灯图形，此时图库管理器消失，出现开发系统窗口的画面，同时鼠标变成直角形，移动鼠标到画面上的适当位置，单击即将选择的指示灯放置到画面上了。同时还可以用鼠标来任意的移动、缩放图形。

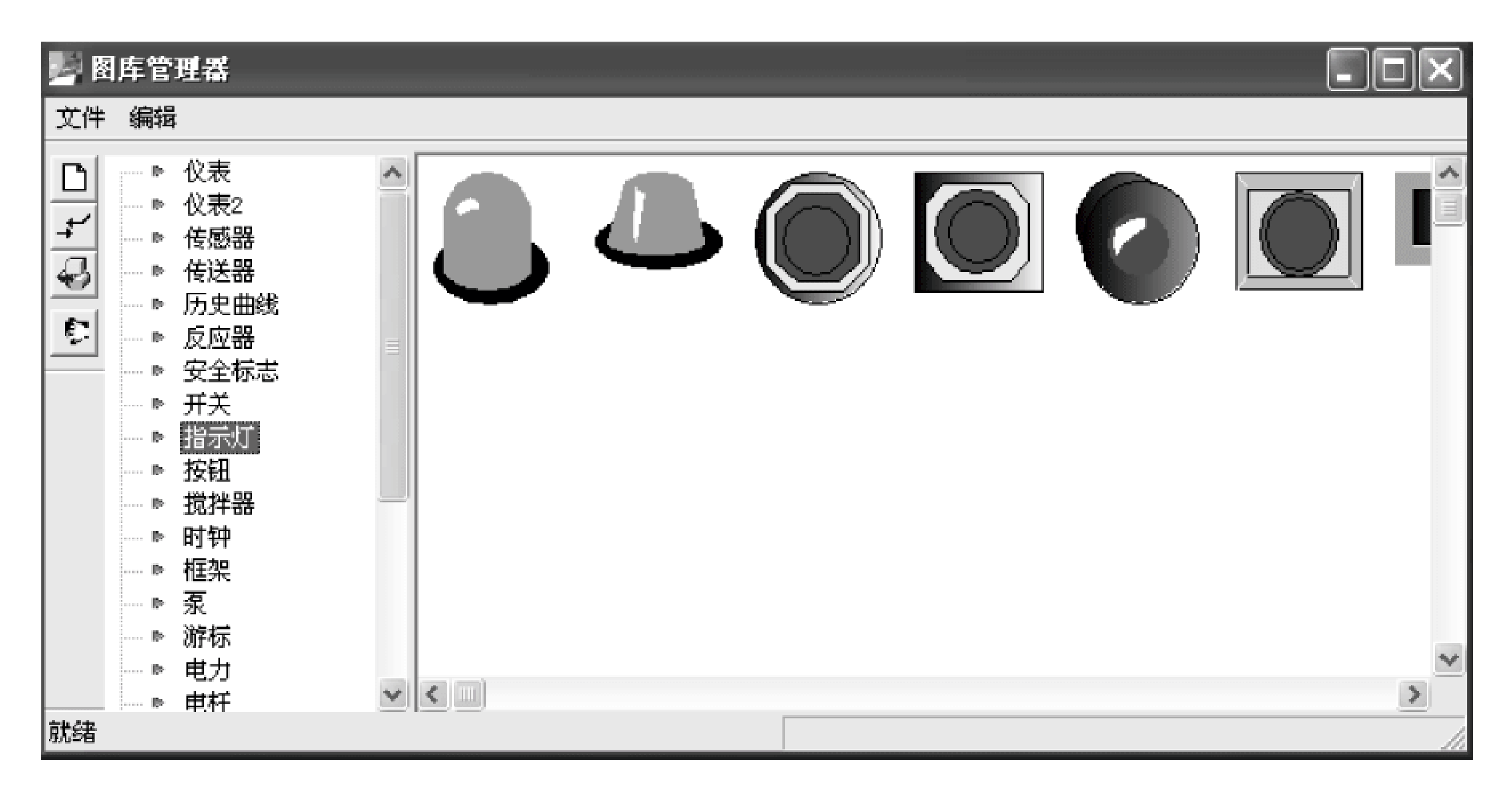


图 9.8　“图库管理器”界面

图库是指组态王中提供的已制作成型的图素组合，图库中的每个成员称为“图库精灵”。

使用图库开发用户工程界面至少有 3 方面的优点。一是降低了工程人员设计组态界面的难度，能更加集中精力于维护数据库和增强软件内部的逻辑控制，缩短开发周期；二是用图库开发的软件具有统一的外观，方便工程人员互相学习交流和掌握；三是利用图库的开放性，工程人员可以生成自己的图库元素，“一次构造，随处使用”，节省了工程人员投资。

组态王为了方便用户更好地管理使用图库，提供了图库管理器，图库管理器集成了图库管理的操作，在统一的界面上，完成“新建图库”、“更改图库名称”、“加载用户开发的精灵”和“删除图库精灵”等操作。

图库精灵中的大部分都有连接向导或者精灵设置外观，双击画面上放置的图库精灵，弹出改变图像外观和定义动画连接的向导对话框，对话框中包含图库精灵的外观修改、与动作连接的变量等各项设置，可以将精灵和数据词典中的定义变量联系起来，但是也有一些精灵没有动画连接，只能作为普通图片使用。不同的图库精灵，具有不同的属性向导界面。用户只需合理调整各项设置就可以设计出符合自己使用要求的个性化图形。

(3) 在图形画面中加入控制操作的按钮图标。在工具箱中选择“按钮”控件将其添加到画面中，然后选中按钮控件单击，在“字符串替换”对话框输入合适的按钮提示信息。设计的

计数器监控系统画面如图 9.9 所示。

用组态王设计应用程序是以画面为程序单位的，每一个画面对应于系统实际运行时的一个 Windows 窗口。用户可以为每个工程建立数目不限的画面。



图 9.9 计数器监控系统图形界面

9.2.5 定义外部设备和数据库

1. 定义外部设备

组态王系统可以与不同的硬件设备组成上下位机系统，根据不同的硬件设备配置相应的通信驱动程序。组态王把那些需要与之交换数据的设备或程序都作为外部设备。

组态王支撑的外部设备包括：①下位机硬件设备，一般是可编程控制器(PLC)、智能模块、板卡、智能仪表和变频器等。工程人员可以把每一台下位机看作一种设备，不必关心具体的通信协议，只需要在组态王的设备库中选择对应设备的类型，然后按照"设备配置向导"的提示一步步完成安装即可，这个过程其本质就是进行驱动程序的配置，有时也称为设备组态。②其他的 Windows 应用程序，它们之间一般是通过 DDE 通信方式交换数据。③网络上其他的计算机终端。组态王支持的几种通信方式有串口通信、数据采集板、DDE 通信、人机界面卡、网络模块以及 OPC 等。

程序在实际运行中是通过 I/O 设备和下位机交换数据的，组态王 I/O 设备管理在工程浏览器中完成，已经配置好驱动程序的设备统一列在工程浏览器界面下的设备分支中。组态王对设备的管理是通过对逻辑设备名的管理实现的。具体讲就是每一个实际 I/O 设备都必须在组态王中指定一个唯一的逻辑名称，此逻辑设备名就对应着该 I/O 设备的生产厂家、实际设备名称、设备通信方式、设备地址、与上位 PC 的通信方式等信息。在组态王中，具体 I/O 设备与逻辑设备名是一一对应的，有一个 I/O 设备就必须指定一个唯一的逻辑设备名，特别是设备型号完全相同的多台 I/O 设备，也要指定不同的逻辑设备名。

只有在定义了外部设备之后，组态王才能通过 I/O 变量和它们交换数据。为方便定义

外部设备，组态王设计了“设备配置向导”引导用户一步步完成设备的连接。当组态王的程序在调试时，可以使用仿真I/O设备模拟下位机向画面程序提供数据，为画面程序的调试提供方便。下面通过组态王提供的仿真PLC和组态王通信建立的过程来介绍外部I/O设备的定义方法。仿真PLC可以模拟实际PLC设备为组态王提供数据，假设仿真PLC连接在计算机的COM1口。

(1) 在组态王工程浏览器左侧选择大纲项“设备\COM1”，在工程浏览器右侧内容显示区双击“新建”图标，运行“设备配置向导”，如图9.10所示。



图9.10　设备配置向导一

(2) 选择“PLC\亚控\仿真PLC”的“COM”项，单击“下一步”，弹出“设备配置向导”，如图9.11所示。



图9.11　设备逻辑名称对话框

（3）为外部设备取一个名称，输入“仿真 PLC”，单击“下一步”，弹出“设备配置向导”，如图 9.12 所示。其中组态王定义的逻辑设备名是由工程人员自己确定，而不一定是实际的设备名称。组态王中的 I/O 变量与具体 I/O 设备的数据交换就是通过逻辑设备名来实现的，当工程人员在组态王中定义 I/O 变量属性时，就要指定与该 I/O 变量进行数据交换的逻辑设备名。



图 9.12　设备通信端口对话框

（4）为设备选择连接串口，假设为 COM1，单击“下一步”按钮，弹出“设备配置向导”，如图 9.13 所示。



图 9.13　设备地址对话框

（5）填写设备地址，假设为 0，单击“下一步”按钮，弹出“通信参数”对话框，如图 9.14 所示。

（6）设置通信故障恢复参数（一般情况下使用系统默认设置即可），单击“下一步”按钮，弹出“设备安装向导”，如图 9.15 所示。



图 9.14　通信参数设置对话框



图 9.15　设备设置完毕对话框

(7) 检查各项设置是否正确，确认无误后，单击“完成”按钮即可。

设备定义完成后，可以在工程浏览器的右侧看到新建的外部设备“仿真 PLC”。在定义数据库变量时，只要把 I/O 变量连接到这台设备上，它就可以和组态王交换数据了。

2. 数据库与变量定义

数据库是“组态王”软件的最核心部分，工业现场的生产状况要以动画的形式反映在屏幕上，操作者在计算机前发布的指令也要迅速送达生产现场，所有这一切都是以实时数据库为中介环节，所以说数据库是联系上位机和下位机的桥梁。在数据库中存放的是变量的当前值。在定义变量时要指定变量名和变量类型，某些类型的变量还需要一些附加信息。数

据库中变量的集合形象地称为“数据词典”,“数据词典”记录了所有用户可使用的数据变量的详细信息。

1) 变量的分类

“数据词典”中存放的是应用工程中定义的变量以及系统变量。组态王系统中定义的变量与一般程序设计语言(比如 Basic、Pascal、C 语言)定义的变量有很大的不同,既要满足程序设计的一般需要,又要考虑到工业控制系统软件的特殊需要。

变量可分为基本类型和特殊类型两大类,基本类型的变量又分为内存变量和 I/O 变量两类。

(1) 内存变量指那些不需要和外部设备或其他应用程序交换数据,也不需要从下位机得到数据,只在组态王内需要使用的变量,比如计算过程的中间变量,就可以设置成“内存变量”。

(2) I/O 变量指的是组态王与外部设备或其他应用程序交换的变量,是工业控制过程中的各种实际物理量。这种数据交换是双向的、动态的,就是说在组态王系统运行过程中,每当 I/O 变量的值改变时,该值就会自动写入外部设备或其他应用程序;每当外部设备或其他应用程序中的值改变时,组态王系统中的变量值也会自动改变。所以,那些从下位机采集来的数据、发送给下位机的指令,比如锅炉液位、电源开关等变量,都需要设置成 I/O 变量。

(3) 特殊变量类型有报警窗口变量、历史趋势曲线变量、系统预设变量 3 种。这几种特殊类型的变量正是体现了“组态王”系统面向工控软件、自动生成人机接口的特色。

2) 变量的数据类型

组态王中变量的数据类型与一般程序设计语言中的变量类型相类似,主要有下面几种。

(1) 实型变量:类似于一般程序设计语言中的浮点型变量,用于表示浮点(float)型数据,取值范围为 10E－38～10E＋38,有效值为 7 位。

(2) 离散变量:类似于一般程序设计语言中的布尔(BOOL)变量,只有 0 和 1 两种取值,常用来表示一些开关量。

(3) 字符串型变量:类似于一般程序设计语言中的字符串变量,主要用于记录一些有特定含义的字符串,比如用户名称、密码等,该类型变量可以进行比较运算和赋值运算。字符串长度最大有效值为 128 个字符。

(4) 整数变量:类似于一般程序设计语言中的有符号长整数型变量,用于表示带符号的整型数据,其取值范围为－2147483648～2147483647。

(5) 结构变量:当组态王工程中定义了结构变量时,在变量类型的下拉列表框中会自动列出已定义的结构变量,一个结构变量作为一种变量类型,结构变量下可包含多个成员,每一个成员就是一个基本变量,成员类型可以为内存离散、内存整型、内存实型、内存字符串、I/O 离散、I/O 整型、I/O 实型、I/O 字符串。

3) 变量的定义

(1) 选择工程浏览器左侧大纲项“数据库\数据词典”,在工程浏览器右侧双击“新建”图标,弹出“定义变量”对话框如图 9.16 所示。

此对话框可以对数据变量完成定义、修改以及数据库的管理工作等操作。在“变量名”处输入变量名,比如“计数器当前值”;在“变量类型”处选择变量类型,比如“内存整数”,其他属性可以不用更改,单击“确定”按钮,则在数据词典中增加一个内存整数变量“计数器当前值”。其他属性的设置详见组态王帮助手册。

注意：如果需要绘制所定义的变量的历史曲线，要在“定义变量”的对话框中通过“记录和安全区”选项卡选择“数据变化记录”或“定时记录”选项，才能在历史趋势曲线中显示此变量的变化情况，这是因为历史趋势曲线中的数据都取自历史数据记录文件，而历史数据记录文件只记录那些设置记录属性为有效的变量。



图 9.16　“定义变量”对话框

(2) 定义一个与指示灯关联的内存离散变量。在“定义变量”对话框中输入变量名“指示灯”，变量类型选择“内存离散”，初始值选择“开”，如图 9.17 所示。定义完成后，单击“确定”按钮，则在“数据词典”中增加一个内存离散变量“指示灯”。



图 9.17　定义“指示灯”对话框

系统中用到的每一个变量都需要进行定义，只有经过定义后的变量，才能被系统中的动画连接、命令语言编程等引用。

9.2.6 建立动画连接

工程人员在组态王开发系统中制作的画面都是静态的，要逼真地显示系统的运行状况，必须在变量定义完成后，将画面中的图素和数据库中已经设定的相应变量对应起来，即让画面"动"起来。这要通过"动画连接"来完成。所谓"动画连接"就是建立画面中的图素与数据库变量的对应关系。这样，工业现场的数据，比如温度、液面高度等，当它们发生变化时，通过 I/O 接口，将引起实时数据库中变量的变化，即当变量的值改变时，在画面上以图形对象的动画效果表示出来，或者由软件使用者通过图形对象改变数据变量的值。

动画连接是设计人机接口的一次突破，它把工程人员从重复的图形编程中解放出来，为工程人员提供了标准的工业控制图形界面，并且由可编程的命令语言连接来增强图形界面的功能。图形对象与变量之间有丰富的连接类型，给工程人员设计图形界面提供了极大的方便。图形对象可以按动画连接的要求改变颜色、尺寸、位置、填充百分数等，一个图形对象又可以同时定义多个连接，组合起来会呈现出令人难以想象的图形动画效果。

对于计数器监控系统的动画连接创建过程主要是进入开发系统，鼠标双击画面中的图形对象，在弹出的动画连接对话框中将定义好的"数据词典"中的变量和相应的对象连接起来。

(1) 在图 9.9 的图形画面开发系统中，双击画面中的文本对象"000"，弹出"动画连接"对话框，单击"模拟值输出"按钮，则会弹出"模拟值输出连接"对话框，在"表达式"处输入"\\本站点\计数器当前值"，可以通过键盘直接输入，也可以单击表达式文本框右边的"?"号，在弹出的"选择变量名"浏览器列表中选择前面已经定义好的变量名"计数器当前值"，单击"确定"按钮，文本框中也会出现"\\本站点\计数器当前值"的表达式。并同时设置整数位数为"3"，小数位数为"0"，如图 9.18 所示。单击"确定"按钮回到"动画连接"对话框，再单击"确定"按钮完成文本框的动画设置返回到组态王开发系统。

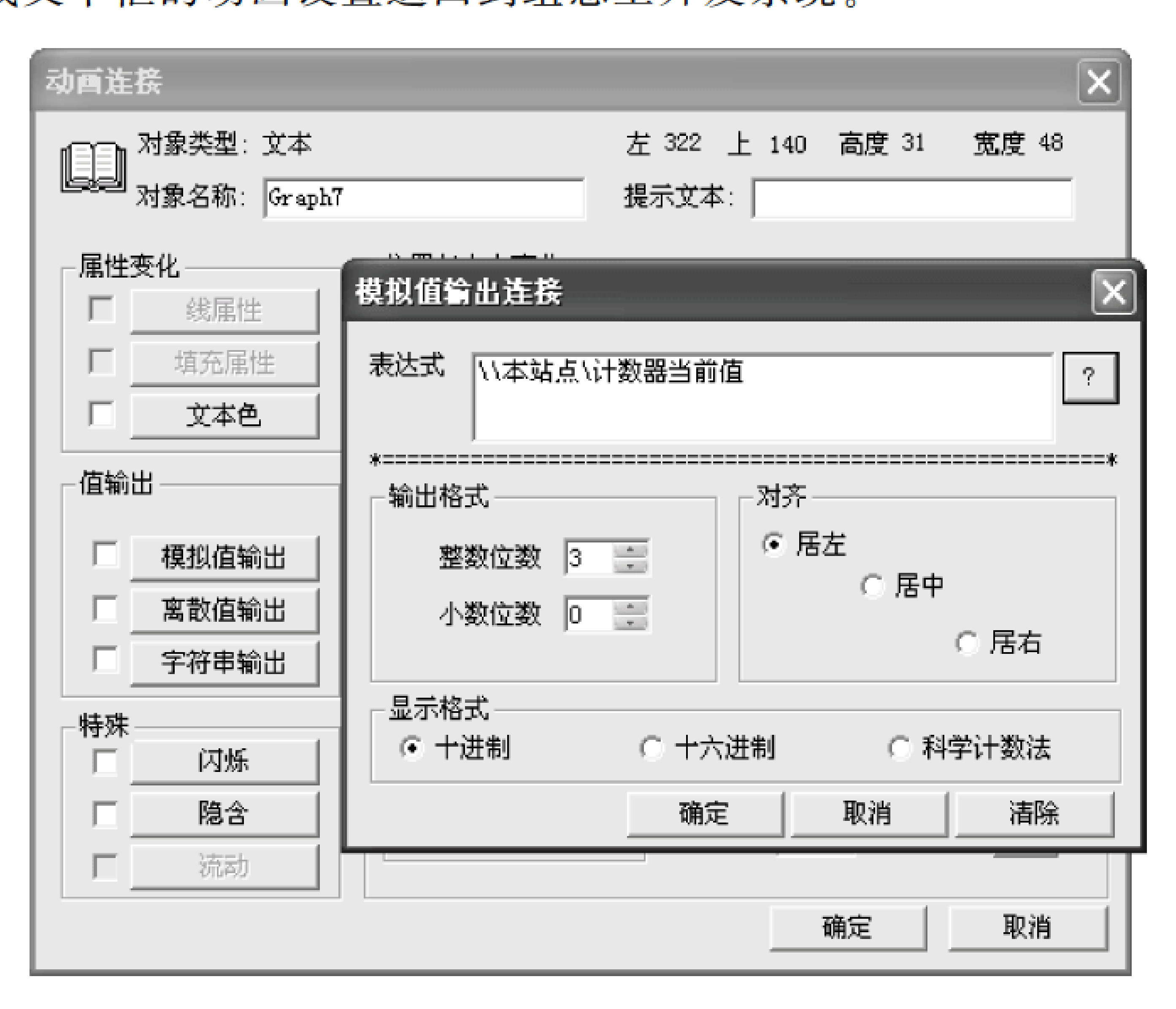


图 9.18 计数器当前值动画设置对话框

（2）在图 9.9 的图形画面开发系统中，双击画面中的“指示灯”对象，在弹出的“指示灯向导”对话框中将变量名设置为“\\本站点\指示灯”，可以直接输入，也可以单击变量名文本框右边的“?”号，在弹出的“选择变量名”浏览器列表中选择前面已经定义好的变量名“指示灯”。根据前面计数器监控系统的要求将指示灯的正常颜色设置为“绿色”，报警色设置为“红色”，如图 9.19 所示。设置完毕后单击“确定”按钮完成指示灯的动画设置返回到组态王开发系统。



图 9.19　指示灯动画设置对话框

（3）在图 9.9 的图形画面开发系统中，双击画面中的“清空”按钮对象，在弹出的“动画连接”对话框中单击“命令语言连接”选项中的“弹起时”按钮，出现“命令语言”窗口，在其下面的编辑栏中输入命令“\\本站点\计数器当前值=0;”，设置完毕后单击“确认”按钮完成“清空”按钮的动画设置返回到组态王开发系统。这样当程序运行时，单击“清空”按钮，程序停止运行并将计数器当前值清空为零。

（4）在图 9.9 的图形画面开发系统中，双击画面中的“关闭”按钮对象，在弹出的“动画连接”对话框中单击“命令语言连接”选项中的“弹起时”按钮，出现“命令语言”窗口，在其下面的编辑栏中输入命令“exit(0);”(exit 函数功能使组态王运行环境退出。其调用形式：exit(Option)；参数 Option：整型变量或数值，其值为 0，退出当前程序；为 1，关机；为 2，重新启动 Windows)，如图 9.20 所示。设置完毕后单击“确认”按钮完成“关闭”按钮的动画设置返回到组态王开发系统。这样当程序运行时，单击“关闭”按钮，程序停止运行并退出。



图 9.20　“关闭”按钮动画设置对话框

在画面组态以及所有的动画连接完成后，必须在开发系统的“文件”菜单下执行“全部存”命令将设计的画面和程序全部保存，对画面所做的任何修改必须保存才有效，没有保存

的画面,在运行时均不会起作用。

9.2.7 命令语言程序编写

在完成画面的动画设置后,还必须通过命令输入,才能控制计数器当前值的变化。各种命令语言通过“命令语言”对话框编辑输入,在组态王运行系统中被编译执行。

在组态王工程浏览器左侧的工程目录显示区中双击“文件”大纲下面“命令语言”下的“应用程序命令语言”项,进入“应用程序命令语言”对话框。单击“运行时”选项卡,将循环执行时间设定为2000ms,然后在命令语言编辑框内输入响应的控制命令,如图9.21所示。注意,命令输入要求在语句的尾部加分号。输入程序时,各种符号都应在英文输入法状态下完成。输入完毕后单击“确认”按钮,完成命令语言的输入。在运行组态王工程时就可以按照指定的命令执行了。注意,对于外部变量的变化,不需要人为地编写命令语言来进行控制。

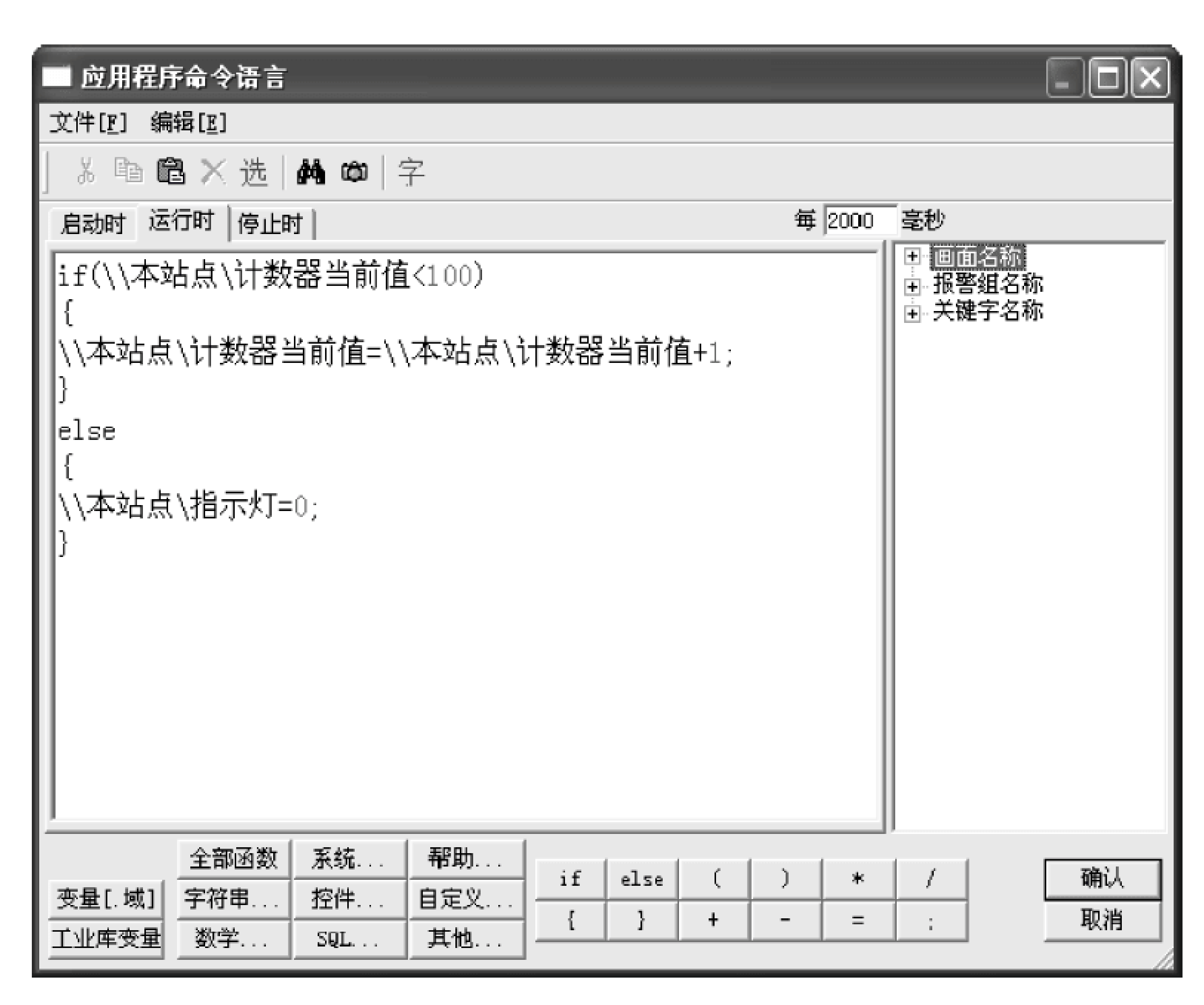


图 9.21 应用程序命令应用编辑栏

9.2.8 组态王运行系统

在完成组态王工程的人机界面组态等工作之后,组态王工程已经初步建立,进入到运行和调试阶段。一般,在组态系统设计上只进行一次是很难开发出令人非常满意的界面,所以在使用组态软件开发后,还必须经过反复的调试修改之后才能达到理想的效果。在完成设计后,组态王系统就可以与实际的设备通信,实现需要的监控要求。

在运行组态王工程之前首先要对运行系统环境进行配置。单击工程浏览器“工程目录显示区”→“系统配置”→“设置运行系统”按钮后或单击快捷工具栏上的“运行”按钮,弹出“运行系统设置”对话框,如图9.22所示。



图 9.22　主画面配置

单击“主画面配置”选项卡，其规定画面运行系统启动时自动调入的画面，如果几个画面互相重叠，最后调入的画面在前面。属性页画面列表对话框中列出了当前应用程序所有有效的画面，选中前面制作的“计数器监控系统”图形画面，单击“确定”按钮即将其配置成主画面。

配置好运行系统后就可以启动运行系统。在开发系统中执行“文件\切换到 View”命令或者单击工程浏览器快捷工具栏上的“VIEW”按钮，即可进入组态王运行系统，如图 9.23 所示。运行后会自动进入“计数器监控系统”画面，无需再进行画面的选择。此时，计数器当前值每隔 2s 加 1，指示灯颜色为绿色，当计数器值变到 100 时指示灯颜色变为红色，单击“清空”按钮可以清空当前计数器值，单击“关闭”按钮退出运行监控系统程序，初步满足了计数器监控系统的设计要求。



图 9.23　计数器监控系统运行画面

如果在运行中有异常，应将系统退回到组态王工程浏览器或者开发系统中作相应的修改，直到系统满足要求工作，完全正常。

如果系统中制作了多个画面，在运行过程中若要切换到其他画面，则单击运行系统菜单"画面\打开"命令，弹出"打开画面"对话框，对话框中列出当前路径下所有未打开画面的清单。选择想要显示的画面的名称，单击"确定"按钮，画面就可以切换到所选择的画面。或单击"取消"按钮撤销当前操作。

9.2.9 组态王信息窗口

组态王信息窗口中显示的信息可以作为一个文件保存在指定的目录中，或者用打印机打印出来，供用户随时查阅。

一般情况下启动组态王系统后，在信息窗口中可以显示的信息主要有组态王系统的启动、关闭、运行模式；历史记录的启动、关闭；I/O设备的启动、关闭；网络连接的状态；与设备连接的状态；命令语言中函数未执行成功的出错信息。

如果用户想要查看与下位设备通信的信息，可以选择运行系统"调试"菜单下的"读成功"、"读失败"、"写成功"、"写失败"等相应选项，则I/O变量读取设备上的数据是否成功的信息也会在信息窗口中显示出来。

组态王的信息窗口如图9.24所示。

用户可以将信息窗口中的信息以"＊.kvl"文件的形式保存到硬盘中，供以后需要时查阅。单击"信息"菜单下"设置存储路径"命令，弹出"设置存储路径"对话框，如果是第1次运行信息窗口，默认保存路径为本机的临时目录"C:\Windows\Temp\"，用户可根据需要点击"浏览…"更改保存路径。一旦用户设置了新的路径后，信息窗口会自动在该路径下生成新的信息文件，以后生成的信息文件都保存到该路径下。默认的信息文件命名方式为"年月日时分.kvl"。



图9.24 组态王信息窗口

9.2.10 组态王系统安全管理

系统的安全保护是应用系统不可忽视的问题，对于可能有不同类型的用户共同使用的大型复杂系统，必须解决好授权与安全性的问题，系统必须能够依据用户的使用权限允许或

禁止其对系统进行操作。组态王提供一个强有力的先进的基于用户的安全管理系统。在组态王系统的开发系统里可以对工程进行加密，打开工程时只有输入密码正确才能进入该工程的开发系统。

1. 组态王开发系统安全管理——工程加密

为了防止其他人员对工程进行修改，在组态王开发系统中可以分别对多个工程进行加密。当进入一个有密码的工程时，只有输入的密码正确时才能进入该工程的开发系统，否则不能打开该工程进行修改，但不会影响工程的运行，从而实现了组态王开发系统的安全管理，同时也保护了开发者的权益。

新建组态王工程，首次进入组态王浏览器，系统默认没有密码，可直接进入组态王开发系统。如果要对该工程的开发系统进行加密，执行工程浏览器中“工具”→“工程加密”命令。弹出“工程加密处理”对话框，如图9.25所示。

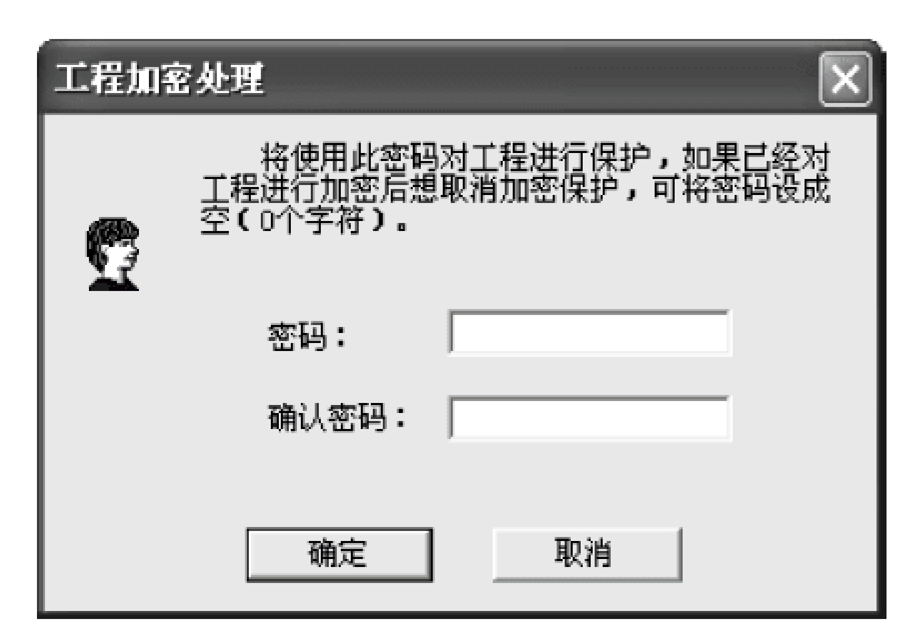


图9.25　“工程加密处理”对话框

在密码栏输入密码，密码长度不超过12B，密码可以是字母（区分字母大小写）、数字、其他符号等，并且需要再次输入相同密码进行确认。

输入完毕后，单击“取消”按钮将取消对工程实施加密操作；单击“确定”按钮后，系统将对工程进行加密。加密过程中系统会弹出提示信息框，显示对每一个画面分别进行加密处理。当加密操作完成后，系统会弹出“操作完成”提示框。

2. 组态王运行系统安全管理

在组态王系统中，为了保证运行系统的安全运行，对画面上的图形对象设置访问权限，同时给操作者分配访问优先级和安全区，当操作者的优先级小于对象的访问优先级或不在对象的访问安全区内时，该对象为不可访问，即要访问一个有权限设置的对象，要求先具有访问优先级，而且操作者的操作安全区必须在对象的安全区内时，才能访问。操作者的操作优先级级别为1～999，每个操作者和对象的操作优先级级别只有一个。系统安全区共有64个，在进行配置时，每个用户可选择除“无”以外的多个安全区，即一个用户可有多个安全区权限，每个对象也可有多个安全区权限。除“无”以外的安全区名称可由用户按照自己的需要进行修改。在软件运行过程中，优先级大于900的用户还可以配置其他操作者，为他们设置用户名、口令、访问优先级和安全区。

9.3　组态王曲线、控件与报表的应用

9.3.1　组态王曲线介绍

组态王系统运行的实时数据和历史数据除了在画面中以值输出的方式和以报表形式显示外，还可以用形象的曲线形式显示。组态王的曲线有趋势曲线、温控曲线和超级 *X-Y*

曲线。

作为实时监控的组态软件，趋势分析是控制软件必不可少的功能，组态王对该功能提供了强有力的支持和简单的控制方法。趋势曲线用于实时显示数据的实时趋势曲线和能够对数据库中的数据进行指定趋势分析的历史趋势曲线两种。这两种曲线外形都类似于坐标值，*X* 轴代表时间，*Y* 轴代表变量值。对于实时趋势曲线最多可显示 4 条曲线，历史趋势曲线最多可显示 16 条曲线，而一个画面中可定义数量不限的趋势曲线（实时趋势曲线或历史趋势曲线）。在趋势曲线中工程人员可以规定时间间距、数据的数值范围、网格分辨率、时间坐标数目、数值坐标数目以及绘制曲线的“笔”的颜色属性等。画面程序运行时，实时趋势曲线可以随时间变化自动卷动，以快速反应变量随时间的变化，但是不能随时间轴回卷，不能查阅变量的历史数据。历史趋势曲线不能自动卷动，它一般与功能按钮一起工作，共同完成历史数据的查看工作。这些按钮可以完成翻页、设定时间参数、启动和停止记录、打印曲线图等复杂功能。

温控曲线反映出实际测量值按设定曲线变化的情况。在温控曲线中，纵轴代表温度值，横轴对应时间的变化，同时将每一个温度采样点显示在曲线中，运行环境中还提供左右两个游标，当工程人员把游标放在某一个温度的采样点上时，该采样点的注释值就可以显示出来。主要适用于温度控制、流量控制等。

超级 *X*-*Y* 曲线主要是用曲线来显示两个变量之间的运行关系，例如电流-转速曲线等，同时也支持多 *Y* 轴曲线。

9.3.2 组态王趋势曲线

趋势曲线用来反映数据变量随时间的变化情况，组态王的图库中有设定好的各种功能按钮的趋势曲线。用户只需定义几个相关变量，适当调整曲线外观，就可以完成曲线指定的复杂功能。

1. 实时曲线

(1) 实时曲线的制作。在组态王开发系统中制作画面时，选择菜单“工具”→“实时趋势曲线”项或单击工具箱中的“实时趋势曲线”按钮，此时鼠标在画面中变为“十”字形，在画面中用鼠标画出一个矩形，实时趋势曲线就在这个矩形中绘出，如图 9.26 所示。

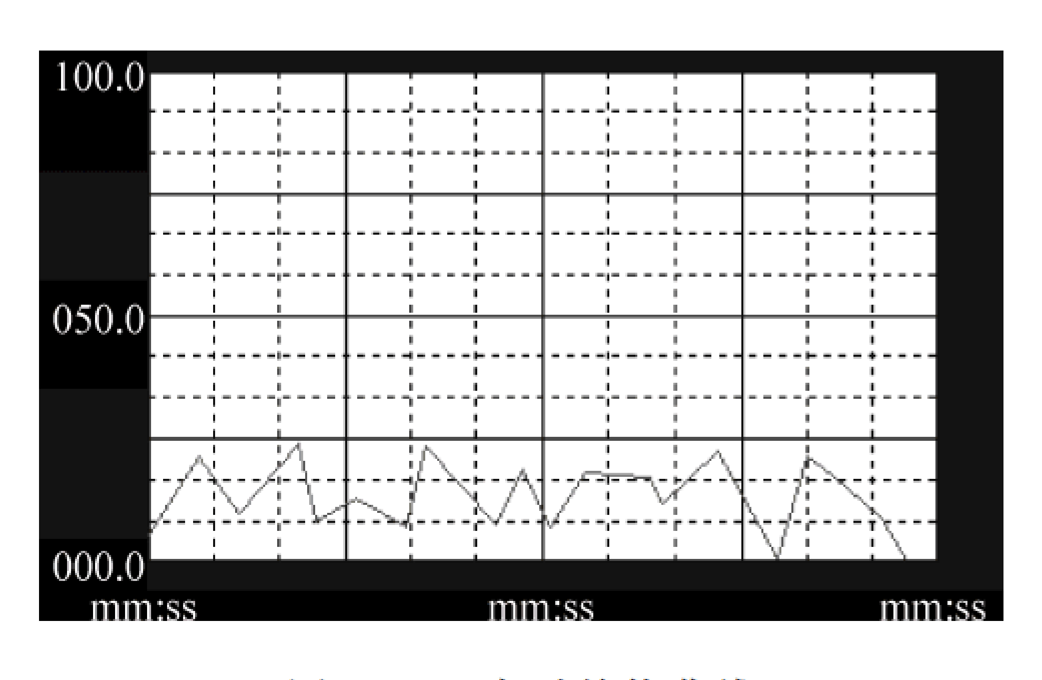


图 9.26 实时趋势曲线

实时趋势曲线对象的中间有一个带有网格的绘图区域，表示曲线将在这个区域中绘出，网格左方和下方分别是 Y 轴(数值轴)和 X 轴(时间轴)的坐标标注。可以通过选中实时趋势曲线对象(周围出现 8 个小矩形)来移动位置或改变大小。在画面运行时实时趋势曲线对象由系统自动更新。

(2) 实时曲线参数的设置。在画面上生成实时趋势曲线对象后，双击创建的实时趋势曲线对象，弹出“实时趋势曲线”属性对话框，如图 9.27 所示。



图 9.27　“实时趋势曲线”对话框

在“曲线定义”选项卡中可以设置包括坐标轴、分割线为短线、边框色、背景色和曲线等内容选项。其中“曲线”项用来定义所绘的 1～4 条曲线 Y 坐标对应的表达式，实时趋势曲线可以实时计算表达式的值，所以它可以使用表达式。实时趋势曲线名的编辑框中可输入有效的变量名或表达式，表达式中所用变量必须是数据库中已定义的变量。右边的“?”按钮可列出数据库中已定义的变量或变量域来供选择。每条曲线可通过右边的线型和颜色按钮来改变线型和颜色。在定义曲线属性时，至少需要定义一条曲线变量。

“标识定义”选项卡中属性设置参考组态王的帮助文档。

2. 历史曲线

组态王提供 3 种形式的历史趋势曲线供用户使用。

第 1 种是从图库中调用已经定义好各功能按钮的历史趋势曲线，对于这种历史趋势曲线，用户只需要定义几个相关变量，适当调整曲线外观即可完成历史趋势曲线的复杂功能，这种形式使用简单方便；该曲线控件最多可以绘制 8 条曲线，但该曲线控件无法实现曲线打印功能。

第 2 种是调用历史趋势曲线控件，这种历史趋势曲线功能很强大，使用比较简单。通过该控件，不但可以实现组态王历史数据的曲线绘制，还可以实现工业库中历史数据的曲线绘制、ODBC 数据库中记录数据的曲线绘制，而且在运行状态下，可以实现在线动态增加/删除

曲线、曲线图表的无级缩放、曲线的动态比较、曲线的打印等。

第 3 种是从工具箱中调用历史趋势曲线，对于这种历史趋势曲线，用户需要对曲线的各个操作按钮进行定义，即建立命令语言连接才能操作历史曲线，对于这种形式，用户使用时自主性较强，能做出个性化的历史趋势曲线；该曲线控件最多可以绘制 8 条曲线，该曲线无法实现曲线打印功能。

无论使用哪一种历史趋势曲线，都需要进行相关配置，主要包括变量属性配置和历史数据文件存放位置配置。

历史曲线的使用和配置详见《组态王用户手册》。

9.3.3 组态王控件的分析

1. 控件的概念

控件实际上是可重用对象，用来执行专门的任务。每个控件实质上都是一个微型程序，但不是一个独立的应用程序，通过控件的属性、方法等控制控件的外观和行为，接收输入并提供输出。例如，Windows 操作系统中的组合列表框就是一个控件，通过设置属性可以决定组合列表框的大小，要显示文本的字体类型，以及显示的颜色。组态王的控件（如棒图、温控曲线、*X*-*Y* 轴曲线）就是微型程序，它们能提供各种属性和丰富的命令语言函数用来完成各种特定的功能。

2. 控件的功能

控件在外观上类似于组合图素，工程人员只需把它放在画面上，然后配置控件的属性，进行相应的函数连接，控件就能完成复杂的功能。

当所实现的功能由主程序完成时需要制作很复杂的命令语言，或根本无法完成时，可以采用控件。主程序只需要向控件提供输入，而剩下的复杂工作由控件去完成，主程序无需理睬其过程，只要控件提供所需要的结果输出即可。另外，控件的可重用性也非常方便。比如画面上需要多个二维条图，用以表示不同变量的变化情况，如果没有棒图控件，则首先要利用工具箱绘制多个长方形框，然后将它们分别进行填充连接，每一个变量对应一个长方形框，最后再把这些复杂的步骤合在一起，才能完成棒图控件的功能。而直接利用棒图控件，工程人员只要把棒图控件复制到画面上，对它进行相应的属性设置和命令语言函数的连接，就可实现用二维条图或三维条图来显示多个不同变量的变化情况。

组态王本身提供很多内置控件，如列表框、选项按钮、棒图、温控曲线和视频控件等，这些控件只能通过组态王主程序来调用，其他程序无法使用，这些控件的使用主要是通过组态王相应控件函数或与之连接的变量实现的。总之，使用控件极大地提高工程人员工程开发和工程运行的效率。

随着 ActiveX 技术的应用，ActiveX 控件也普遍被使用。组态王支持符合其数据类型的 ActiveX 标准控件。这些控件包括 Microsoft Windows 标准控件和任何用户制作的标准 ActiveX 控件。这些控件在组态王中称为“通用控件”，在组态王程序中凡提到“通用控件”，即是指 ActiveX 控件。

3. 组态王内置控件

组态王内置控件是组态王提供的，只能在组态王程序内使用的控件。它能实现控件的功能，组态王通过内置的控件函数和连接的变量来操作、控制控件，从控件获得输出结果。其他用户程序无法调用组态王内置控件。这些内置控件包括棒图控件、温控曲线、*X*-*Y* 曲线、列表框、选项按钮、文本框、超级文本框、AVI 动画播放控件、视频控件、开放式数据库查询控件、历史曲线控件等。

在组态王中加载内置控件，可以单击工具箱中的“插入控件”按钮，或选择画面开发系统中的“编辑\插入控件”菜单。系统弹出“创建控件”对话框，如图 9.28 所示。



图 9.28　“创建控件”对话框

对话框左侧的“种类”列表中列举了内置控件的类型，选中某一项，在右侧的内容显示区中可以看到该类中包含的控件。选择控件图标，单击“创建”按钮，则创建控件；单击“取消”按钮，则取消创建。

4. 控件的应用

以立体棒图控件的制作和应用为例来了解控件的应用，其他控件的制作和使用可参考组态王帮助手册。

棒图是指用图形的变化表现与之关联的数据的变化的绘图图表。组态王中的棒图图形可以是二维条形图、三维条形图或饼图。

(1) 创建棒图控件到画面。使用棒图控件，需要先在画面上创建控件。打开组态王画面开发系统，单击工具箱中的“插入控件”按钮或选择画面开发系统中的“编辑”→“插入控件”菜单。系统弹出如图 9.28 所示的“创建控件”对话框。在种类列表中选择“趋势曲线”，在右侧的内容中选择“立体棒图”图标，单击对话框上的“创建”按钮，或直接双击“立体棒图”图标，关闭对话框。此时鼠标变成小“十”字形，在画面上需要插入控件的地方按下鼠标左键，拖动鼠标，画面上出现一个矩形框，表示创建后控件界面的大小。松开鼠标左键，控件在画面上显示出来，如图 9.29 所示。

控件周围有带箭头的小矩形框，鼠标移动到小矩形框上，鼠标箭头变为方向箭头时，按

下鼠标左键并拖动,可以改变控件的大小。当鼠标在控件上变为双“十”字形时,按下鼠标左键并拖动,可以改变控件的位置。

棒图中的每一个条形图下面对应一个标签 L1、L2、L3、L4、L5、L6。这些标签分别和组态王数据库中的变量相对应,当数据库中的变量发生变化时,则与每个标签相对应的条形图的高度也随之动态地发生变化,因此通过棒图控件可以实时地反映数据库中变量的变化情况。另外,用户还可以使用三维条形图和二维饼形图进行数据的动态显示。

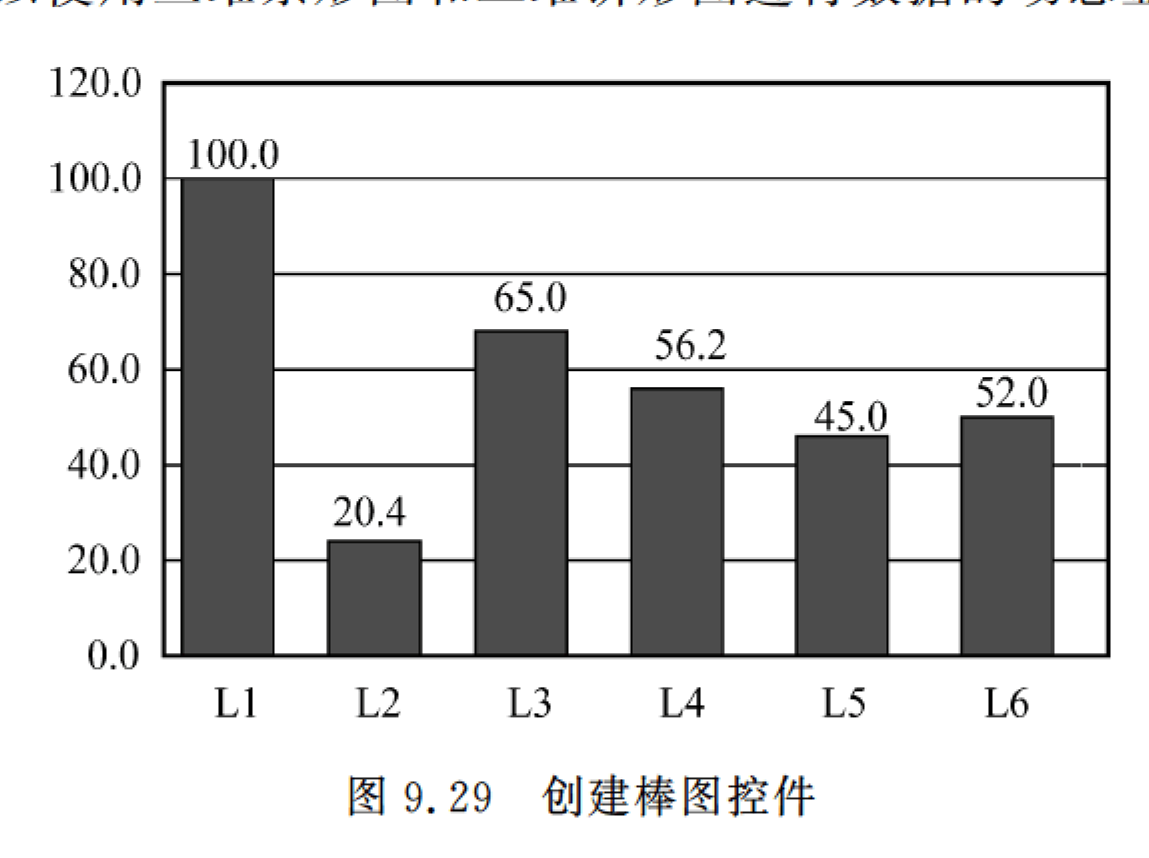


图 9.29 创建棒图控件

(2) 设置棒图控件的属性。双击棒图控件,弹出如图 9.30 所示棒图控件属性设置对话框,此属性页用于设置棒图控件的控件名称、图表类型、标签位置、颜色设置、刻度设置、字体型号、显示属性等各种属性,用户可以根据自己工程的需要进行相应的选择和设置。



图 9.30 棒图属性设置对话框

(3) 棒图控件的动画连接。设置完棒图控件的属性后,就可以准备使用所建立的控件了。棒图控件的动画连接要与变量关联,棒图的刷新都是使用组态王提供的棒图函数来完成的。组态王的棒图函数有以下这些。

chartClear("ControlName"),此函数用于在指定的棒图控件中清除所有的棒形图。

chartAdd("ControlName", Value, "label") ,此函数用于在指定的棒图控件中增加一个新的条形图。

chartSetBarColor("ControlName", barIndex, colorIndex) ,此函数用于在指定的棒图控件中设置条形图的颜色。

chartSetValue("ControlName", Index, Value),此函数用于在指定的棒图控件中设定/修改索引值为 Index 的条形图的数据。

以上 4 个函数中,ControlName 为用户定义的棒图控件名称,可以为中文或者英文名;Value 为设定条形图的初始值,实型数据;label 为设定条形图的标签值。这些函数的具体参数及用法参考《组态王函数速查手册》。

在组态王画面开发系统的画面空白处右击,在弹出的快捷菜单中选择“画面属性”,单击“画面属性”对话框的“命令语言”按钮,在弹出的“画面命令语言”对话框中输入以下程序:

```
chartClear( "计数器数值变化曲线" );
chartAdd( "计数器数值变化曲线" , \\本站点\计数器当前值, "计数器当前值" );
```

单击“文件\全部存”,切换到画面运行系统,就可以看到如图 9.31 所示的计数器当前值的棒图效果。

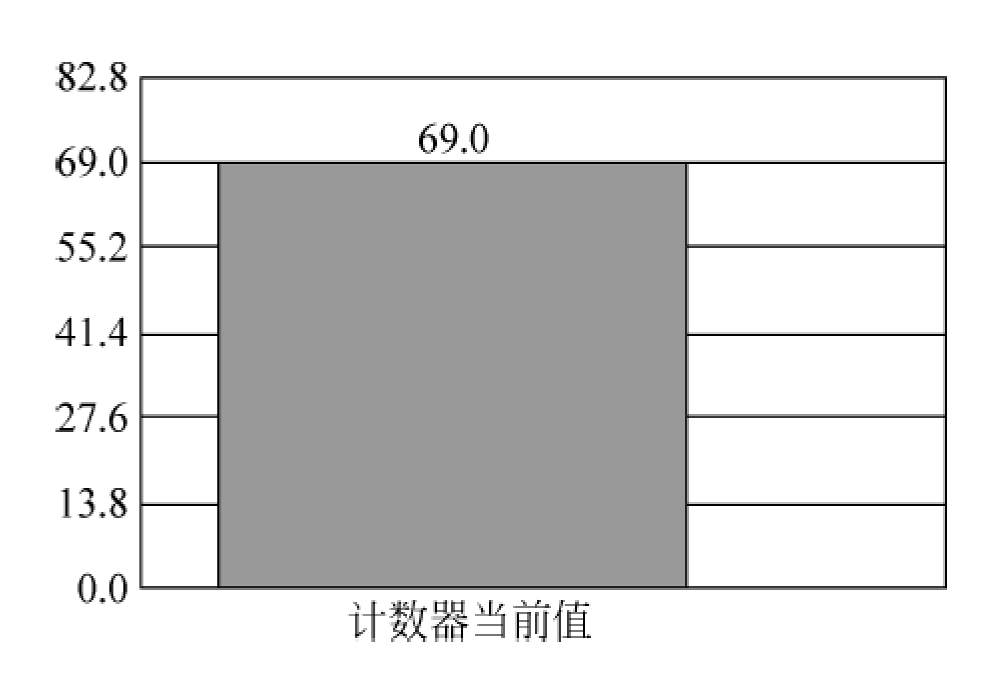


图 9.31　计数器当前值棒图显示画面

9.3.4　组态王报表系统

数据报表是反映生产过程中的数据、状态等,并对数据进行记录的一种重要形式,是生产过程中必不可少的重要部分。报表不仅能反映系统实时的生产情况,也能对长期的生产过程进行统计、分析,从而使管理人员能够实时掌握和分析生产情况。

组态软件除了能够实时显示数据和存储数据外,生成报表的功能也是必不可少的。组态王提供了内嵌式报表系统,工程人员可以任意设置报表格式,对报表进行组态。组态王也为工程人员提供了丰富的报表函数,实现各种运算、数据转换、统计分析、报表打印等功能。既可以制作实时报表,也可以制作历史报表。组态王还支持运行状态下单元格的输入操作,在运行状态下通过鼠标拖动改变行高、列宽。另外,工程人员还可以制作各种报表模板,实现多次使用,以免重复工作。

1. 实时数据报表

(1) 进入组态王开发系统,创建一个新的画面,在组态王工具箱按钮中,单击“报表窗

口”按钮，此时，鼠标箭头变为小“＋”字形，在画面上需要加入报表的位置按下鼠标左键，并拖动，画出一个矩形，松开鼠标左键，报表窗口创建成功，如图 9.32 所示。鼠标箭头移动到报表区域周边，当鼠标形状变为双“＋”字形箭头时，按下左键，可以拖动表格窗口，改变其在画面上的位置。将鼠标移动到报表窗口边缘带箭头的小矩形上，这时鼠标箭头形状变为与小矩形内箭头方向相同，按下鼠标左键并拖动，可以改变报表窗口的大小。当在画面中选中报表窗口时，会自动弹出“报表工具箱”，不选择时，“报表工具箱”自动消失。“报表工具箱”中的工具可以用来设置报表格式、编辑表格中的显示内容等。

(2) 双击报表窗口的灰色部分(表格单元格区域外没有单元格的部分)，弹出“报表设计”对话框，如图 9.33 所示。该对话框主要设置报表的名称、报表表格的行列数目以及选择套用表格的样式。组态王中每个报表窗口都要定义一个唯一的标识名，该标识名的定义应该符合组态王的命名规则，标识名字符串的最大长度为 31。

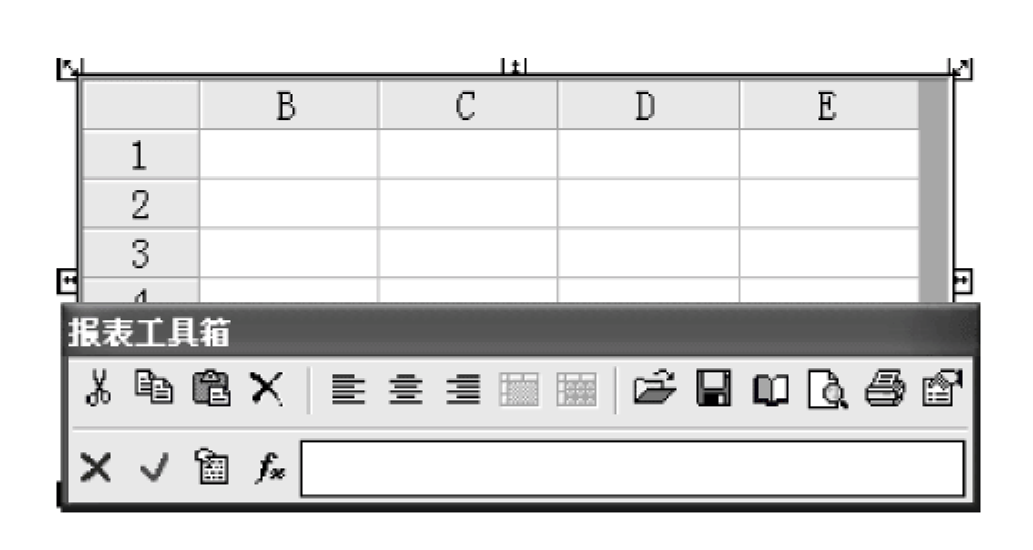


图 9.32 报表窗口



图 9.33 “报表设计”对话框

(3) 与 Word 的表格设计一样需要设计表头。选中要使用的表格，单击报表工具箱上的“合并单元格”按钮，在报表工具箱的编辑框内输入文本“计数器数值实时报表”，单击“输入”按钮，或者双击合并的单元格，使输入光标位于该单元格中，输入上述文本也可一样完成。

(4) 设计报表日期。双击要显示日期的位置，利用组态王系统中预设变量中的 8 个时间变量，在单元格中输入“＝Date($年，$月，$日)”，若要显示当前时间，可以输入“＝Date($时，$分，$秒)”。其中，$年：返回系统当前日期的年份；$月：返回 1～12 之间的整数，表示当前日期的月份；$日：返回 1～31 之间的整数，表示当前日期的日；$时：返回 0～23 之间的整数，表示当前时间的时；$分：返回 0～59 之间的整数，表示当前时间的分数；$秒：返回 0～59 之间的整数，表示当前时间的秒数。

(5) 设计报表具体显示格式。比如要在第 5 行第 1 个单元格中显示“计数器当前值：”，可以在 A5 中直接输入“计数器当前值：”即可。

在运行系统中，报表单元格是不允许直接输入的，所以要使用函数来操作。单元格操作函数是指可以通过命令语言来对报表单元格的内容进行操作，或从单元格获取数据的函数。这些函数大多只能用在命令语言中。比如在运行中要在 B5 中显示计数器当前值，就要先选中 B5 单元格，在组态王的数据改变命令语言中输入以下命令：

```
ReportSetCellValue("计数器数值实时报表",5,2, \\本站点\计数器当前值);
```

其中，ReportSetCellValue 设置单个单元格数值函数，将指定报表的指定单元格设置为给定值。其使用格式为：

```
Long nRet = ReportSetCellValue(String szRptName, long nRow, long nCol, float fValue)
```

返回值：整型 0——成功；1——行列数小于等于零；2——报表名称错误；3——设置值失败。

参数说明：szRptName：报表名称。Row：要设置显示数值的报表的行号(可用变量代替)。Col：要设置显示数值的报表的列号(这里的列号使用数值，也可用变量代替)。Value：要设置的数值。

单元格设置函数还有 ReportSetCellString()、ReportSetCellValue2()、ReportSetCellString2()。这些函数的使用方法详见组态王函数手册。

(6) 保存报表。在开发状态下，单击报表工具箱中的“保存”按钮，选择保存路径，输入要保存的报表文件名，单击“保存”按钮，这样，一个简单的实时数据报表就生成了。保存的报表可供以后需要时调用。

(7) 打印报表。在画面上新建一个“打印报表”按钮，在弹起时的命令语言对话框中输入：ReportPrint2("计数器数值实时报表",0)；

报表打印函数根据用户的需要有两种使用方法，一种是执行函数时自动弹出“打印属性”对话框，供用户选择确定后，再打印；另外一种是执行函数后，按照默认的设置直接输出打印，不弹出“打印属性”对话框，适用于报表的自动打印。报表打印函数原型为：

```
ReportPrint2(String szRptName)
```

或者

```
ReportPrint2(String szRptName, EV_LONG|EV_ANALOG|EV_DISC) ;
```

函数功能：将指定的报表输出到打印配置中指定的打印机上来打印。

参数说明：szRptName：要打印的报表名称。EV_LONG|EV_ANALOG|EV_DISC：整型或实型或离散型的一个参数，当该参数不为 0 时，自动打印，不弹出“打印属性”对话框。如果该参数为 0 时，则弹出“打印属性”对话框。

设计好的实时数据报表界面如图 9.34 所示。

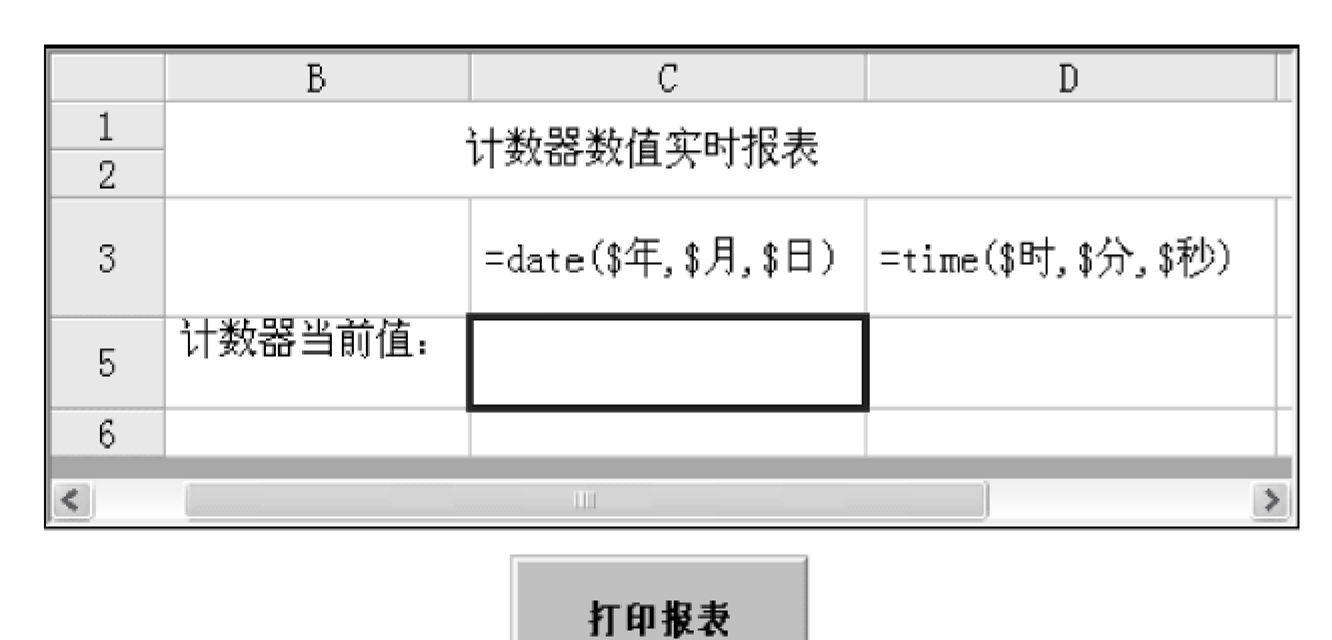


图 9.34　实时数据报表界面

保存画面，运行后的结果如图 9.35 所示。单击“打印报表”，弹出打印设置对话框，设置好后数据报表即可进行打印。



图 9.35　实时数据报表运行界面

2. 历史数据报表

历史报表记录了以往的生产记录数据，对用户是非常重要的。历史报表的制作和实时报表的制作方法都是一样的，还可以通过调用历史报表查询函数加以实现。

在图 9.34 的实时数据报表中添加一个“数据报表查询”按钮，在弹起时的命令语言对话框中输入：ReportSetHistData2()；运行组态王系统，单击“数据报表查询”按钮，弹出“报表历史查询”对话框，如图 9.36 所示。在对话框中输入适当的查询参数值，然后单击“确定”按钮，就可以按照要求查出指定变量在指定时间段的历史数据。



图 9.36　“报表历史查询”对话框

注意：要查询的变量必须是记录历史数据的变量，需要在工程浏览器的数据词典中对定义的变量进行设置，比如针对前面建立的“计数器当前值”变量，双击变量名在弹出的“变量定义”的“记录和安全区”选项卡下选择“定时记录”或者“数据变化记录”，这样设置以后，

才能在图 9.37 的“报表历史查询”对话框的“变量选择”选项卡中的“历史库变量”中，显示需要查询历史数据的变量，如图 9.37 所示。



图 9.37　“变量选择”选项卡

9.4　组态王的网络应用

9.4.1　组态王网络结构概述

组态王完全基于网络的概念，是一种真正的客户机-服务器模式，支持分布式历史数据库和分布式报警系统，可运行在基于 TCP/IP 网络协议的网络上，使用户能够实现上、下位机以及更高层次的厂级连网。

TCP/IP 网络协议提供了在不同硬件体系结构和操作系统的计算机组成的网络上进行通信的能力。一台 PC 通过 TCP/IP 网络协议可以和多个远程计算机（即远程节点）进行通信。

组态王的网络结构是一种柔性结构，可以将整个应用程序分配给多个服务器，可以引用远程站点的变量到本地使用（比如显示、计算等），这样可以提高项目的整体容量结构并改善系统的性能。服务器的分配可以是基于项目中物理设备结构或不同的功能，用户可以根据系统需要设立专门的 I/O 服务器、历史数据服务器、报警服务器、登录服务器和 Web 服务器等。

工程人员要实现组态王的网络功能，必须满足以下两个条件。

（1）将组态王安装在网络版 Windows 98/2000/NT 或 Windows XP 上，并在配置网络时绑定 TCP/IP 协议，即利用组态王网络功能的 PC 必须首先是某个局域网上的站点并启动该网。

（2）客户机和服务器必须安装并同时运行组态王（除 Internet 版本的客户端）软件。

9.4.2 组态王网络配置及应用

要实现组态王的网络功能，除了具备网络硬件设施外，还必须对组态王各个站点进行网络配置，设置网络参数，并且定义在网络上进行数据交换的变量，报警数据和历史数据的存储和引用等。

（1）在组态王工程浏览器中，选择菜单“配置”→“网络设置”，或者在目录显示区中，选择大纲项系统配置下的成员网络配置，双击网络配置图标，弹出“网络配置”对话框，如图 9.38 所示。网络配置有网络参数、节点类型和客户配置 3 个属性页，其具体使用详见组态王说明书。



图 9.38 “网络配置”对话框

组态王运行分单机和联网两种模式，所有进入网络的计算机都要选择“联网”运行模式。本地节点名就是本地计算机名称。进入网络的每一台计算机必须具有唯一的节点名，本地节点名也可以使用本地主网卡的 IP 地址，比如输入电脑 IP 地址：59.69.141.48。同时在“节点类型”选项卡中，选中“本机是登录服务器”、“本机是 I/O 服务器”、“进行历史数据备份”、“本机是校时服务器”、“本机是报警服务器”、“本机是历史记录服务器”，单击“确定”按钮即完成服务器的配置。

（2）网络端口的配置。在进行网络 IE 浏览器访问时，需要知道被访问程序的端口号，在组态王网络访问之前，需要先定义组态王的端口号。

在工程浏览器界面的 Web 目录下双击“发布画面”，弹出如图 9.39 所示的“页面发布向导”配置对话框。

其中站点名称是 Web 服务器名称，自动从系统获取的，不能修改。默认端口就是 IE 浏览器与运行系统进行网络连接的端口号，可按实际情况进行修改。其他选项的设置主要影响浏览器中显示的画面。



图 9.39　"页面发布向导"对话框

(3) 在工程浏览器界面的 Web 目录下选中"发布画面"，双击右边窗口的"新建"图标，弹出如图 9.40 所示的"Web 发布组配置"对话框。在该对话框中可完成发布组名称的定义以及描述、要发布画面的选中以及其他的一些配置。



图 9.40　"Web 发布组配置"对话框

(4) 在实现IE访问浏览前,要先启动组态王运行系统。打开IE浏览器,首先输入访问的组态王运行系统的Web地址。地址格式为:http://发布站点机器名或者IP地址:组态王Web定义端口号。

在浏览器地址栏输入http://59.69.141.48:8001/,回车后即进入如图9.41所示的发布组界面。单击发布组界面上的组名"wangluoyanshi",进入组的浏览界面,网络中所示的画面与组态王运行系统一样的逼真,如图9.42所示。



图9.41 组态王发布组列表界面



图9.42 计数器监控系统网络浏览界面



习题

1. 简述组态软件的基本特点。

2. 简述组态王的基本功能。

3. 简述使用组态王开发应用程序的一般步骤。

4. 使用组态王软件的基本功能,设计一个简单液位监控系统。

5. 使用组态王软件,练习开发一个锅炉温度控制系统,包括主画面、趋势曲线画面和报表画面等。

6. 简述组态王软件网络功能的配置和实现。

第10章 计算机控制系统设计与实现

计算机控制系统的设计，是一个理论问题与工程实际相结合的综合过程，涉及自动控制技术、自动检测技术、计算机技术、通信技术和微电子技术等。其设计一般应从工艺要求、确定控制任务开始，选择控制主机机型，确定控制算法、系统总体设计方案、硬件设计、软件设计，选择传感器、变送器和执行器等工业自动化仪表，系统调试，最后进行生产现场调试，达到所要求的控制性能指标为止。

本章主要介绍计算机控制系统的设计原则与步骤、计算机控制系统的工程设计与实现，并给出了计算机控制系统的设计应用实例。

10.1 系统设计的原则与步骤

由于控制对象是多种多样的，不同的被控对象或生产过程，要求计算机实现的控制功能也各不相同，其控制系统的设计方法和具体技术指标也千变万化，但在计算机控制系统的设计和实施过程中，所遵循的设计原则和设计步骤大致是相同的，这些共同的原则和步骤在设计前或设计过程中都必须予以考虑。

10.1.1 计算机控制系统设计原则

1. 安全可靠

系统的可靠性是指系统在规定的条件下和规定的时间内完成规定功能的能力，它表示系统长期、稳定进行工作的能力。在计算机控制系统中，可靠性指标一般用系统的平均无故障时间（Mean Time Between Failure，MTBF）和平均维修时间（Mean Time to Repair，MTTR）来表示。MTBF 反映了系统可靠工作的能力，MTTR 表示系统出现故障后立即恢复工作的能力，一般希望 MTBF 要大于某个规定值，而 MTTR 值则是越小越好。

对工业控制计算机系统最基本的要求是安全性和可靠性要高，它们也是计算机控制系统设计中最重要的一个要求。这是因为工业生产现场环境比较恶劣，工作现场干扰源多，系统的正常运行总是受到周围各种干扰的威胁，一旦发生故障，轻则影响生产，造成产品质量不合格，带来经济损失；重则造成严重事故，比如人员的伤亡和设备的损坏。所以在计算机控制系统的设计过程中，必须把安全可靠性放在首位，以保证生产安全、可靠和稳定地运行。

为了保证计算机控制系统的高可靠性，在设计过程中应采取各种有利于系统安全可靠

的技术措施和控制方案。

首先是选用高性能的工业控制计算机或者其他的主机模式，保证在比如粉尘、潮湿、振动、冲击、高温、电磁干扰等恶劣的工业环境下，系统仍能长时间正常运行。其次是设计性能可靠的控制方案，并考虑各种安全保护措施，使系统具有异常报警、事故预测、故障诊断与事故处理、安全联锁以及不间断电源等功能。再次通常为了预防计算机故障，还设计有附加后备装置。对于一般的控制回路，选用手动操作器作为后备，对于必须采用自动控制的重要回路或者特殊回路，选用常规控制仪表或者后备计算机作为后备。这样一旦计算机出现故障，就可以把后备装置切换到控制回路中，去维护生产过程的正常运行。

对于特殊的控制对象，可以设计两台计算机互为备用地去执行控制任务，称为双机系统。在双机系统中，用两台计算机作为系统的核心控制器，由于两台计算机都发生故障的概率很低，因此大大提高了系统的可靠性。双机系统中的两台计算机有备份工作方式、主从工作方式和双机工作方式。

在备份工作方式下，一台计算机作为主机投入系统运行，另一台作为备份机处于通电状态，但是它是脱离系统的，只是作为系统的一台备份机。若投入运行的主机出现故障时，通过专用的程序切换装置，自动地将备份机切入系统运行作为主机，承担起主机的任务，而被替换下来的原主机经过故障排除修复后则转作为备份机，处于待命状态。这样不会因为主机故障而影响系统正常运行。

主从工作方式是指两台计算机同时投入系统运行，在正常工作方式下，两台计算机分别执行不同任务，如一台承担整个系统的主要控制任务（主机），另一台则执行一般的数据处理和部分设备的控制工作（从属机）。当其中一台发生故障时，故障机能自动地脱离系统，另一台计算机能自动地承担起系统所有的控制任务，以保证系统的正常运行。

双机工作方式是指两台计算机同时投入运行，且任何一个时刻都同步执行同一个任务，并把两台计算机的执行结果进行比较核对，如果比较相同，则说明两台计算机工作正常，可以输出比较后的结果。如果比较结果不同，则说明其中一台计算机有问题，此时应封锁输出，同时对前一步进行重复处理，然后再核对校验两机结果，若经几次核对后两机结果仍不相符，则启动故障诊断程序，将其中一台故障机从系统中隔离，另一台作为主机继续执行控制任务。

另外控制系统设计中还要有完善的抗干扰措施，硬件抗干扰措施有滤波、接地、屏蔽、隔离、设置干扰吸收网络以及合理布线等，软件抗干扰方法有数字滤波、指令冗余、软件陷阱和看门狗技术等，同时还要考虑电源的抗干扰问题。

2. 实时性强

实时性是计算机控制系统的重要指标之一，表现在对内部和外部事件能及时地响应，并在规定的时间内做出相应的处理，不丢失信息，不延误操作。计算机处理的事件一般分为两类：一类是定时事件，比如数据的定时采集、运算处理和控制等，对于定时事件系统应设置时钟，保证定时处理；另一类是随机事件，比如事故出现后的报警、打印要求等。对于随机事件系统应设置中断，并根据故障的轻重缓急预先分配中断级别，一旦事故发生，以保证优先处理紧急故障。

3. 通用性好

通用性好是指设计的计算机控制系统能根据各种不同的设备和不同控制对象的控制要求,扩充灵活、修改方便。工业计算机控制的对象千变万化,而计算机控制系统的研发又需要一定的投资和周期,因此,不可能为一个生产过程或一台装置研制一台专用计算机,常常是设计或者选用通用性好的计算机控制装置灵活地构成系统。当设备和控制对象发生变化时,或者再设计另外一套控制系统时,通用性好的系统一般是稍作更改或扩充就可以适应。

尽管控制对象多种多样,但从控制功能来分析和归类,仍然可以找到许多共性。因此系统设计时应尽量考虑能适应这些共性,尽可能地采用标准化设计,以及积木式的结构,在此基础上,再根据各种不同设备和不同控制对象的控制要求,灵活地构造系统。

计算机控制系统的通用灵活性主要体现在两个方面:一是硬件设计方面首先采用标准总线结构,配置各种通用的功能模板,以便在扩充功能时,只需增加功能模板即可;二是软件方面,应采用标准模块结构,用户在使用时不需要二次开发,只需按控制要求选择各种功能模块,灵活地进行控制系统的组态。

同时在系统设计中,各项设计指标最好留有一定的余量,这也是可扩充的首要条件。比如计算机的工作处理速度,如果在设计时不留有一定的余量,那么想再进行系统的扩充是完全不可能的。其他的比如电源功率、存储容量等也应留有一定的余量。

4. 操作性能好,维护维修方便

对于一个计算机控制系统来说,所谓操作性好就是指系统有良好的人机界面,方便操作、运行,便于掌握,并不强求操作工要掌握计算机知识才能操作。在设计时既要考虑操作的先进性,又要兼顾操作人员原有的习惯,降低对操作人员专业知识的要求。在硬件配置方面应考虑控制开关不能太多太复杂,操作顺序要尽量简单,控制台要便于操作人员工作,人机界面可以采用 CRT、LCD 或者触摸屏等,使操作人员对现场情况一目了然。

维护维修方便是指一旦发生故障,易于查找和排除故障。这需要从硬件和软件两个方面结合起来考虑。硬件上宜采用标准的功能模板式结构,便于及时更换故障模板,同时模板上应有工作状态指示灯和监测点,便于维修人员检查。软件上要配置有差错程序和诊断程序,以便发生故障时通过程序帮助来查找故障发生的部位,缩短排除故障的时间。必要时还要考虑容错程序,以便在发生故障时能保证系统处于安全的状态。

5. 性能价格比高,经济效益高

计算机控制系统的设计在满足系统生产工艺所要求的性能指标的前提下,要有市场竞争意识,应尽可能地降低成本,带来良好的经济效益。经济效益主要体现在两个方面:一是系统设计的性能价格比要尽可能的高,要有一定的预见性,尽可能的缩短设计周期,投入产出比尽可能的低;二是提高产品的质量与产量、降低能耗、减少污染、改善生产条件等经济和社会效益各方面来进行综合评估。

10.1.2 计算机控制系统设计步骤

计算机控制系统的设计虽然随着被控对象、控制方式和系统规模的不同而有所差异,但

系统设计与实施的基本内容和主要步骤大致相同。如果设计步骤不清晰，或者每一步需要做什么不明确，就有可能引起设计过程中的混乱甚至返工。作为一个计算机控制系统工程项目，其设计研制过程中可划分为4个阶段：准备阶段（确定任务阶段）、工程设计阶段、离线仿真和调试阶段以及在线调试和投运阶段。

1. 确定任务阶段

随着市场经济的发展以及规范化，计算机控制系统项目的设计与实施过程中，通常存在着甲方乙方关系。所谓甲方通常是指项目的需求方或者是任务的委托方，可能是用户本身、上级主管部门或者中介单位；乙方通常是计算机控制系统项目的承建实施方。无论是甲方或者乙方，都需要对整个工程项目与控制任务的确定有所了解。在一个计算机控制系统项目的研制过程中，确定任务阶段一般按照下面流程来进行。

(1) 甲方提出项目任务委托书。在委托乙方承接系统工程项目前，甲方一般要给出正式的书面任务委托书，其中委托书中一定要提供明确的系统技术性能指标要求，还要包括经费、计划进度、合作方式等相关内容。

(2) 乙方研究其任务委托书。乙方在接到甲方的书面任务委托书后逐条进行认真研究，对含义不清、认识上有分歧、需要补充或删减的地方逐条标出，并拟订需要进一步讨论的问题与修改的意见。

(3) 双方对委托书进行确认性修改。在乙方对任务委托书进行了认真的研究之后，甲乙双方应就委托书的内容进行协商性的讨论、修改进而最终确认。为避免因行业和专业不同所带来的局限性，讨论时应有各方面有经验的人员参加。经过确认或修改后的委托书中不应再有含义不清的词汇与条款，并且一定要明确双方的任务和技术工作界面。

(4) 乙方初步进行项目系统总体设计方案。由于项目实施的任务和经费没有具体落实，所以这个阶段的总体设计方案只是粗线条的。如果条件允许，应多做几个方案以便进行比较。方案中应突出技术难点及解决办法、经费概算和预计完成时间等关键问题。

(5) 乙方进行方案可行性论证。方案可行性论证的目的是要估计承接该项任务的把握性，并为签订合同后的设计工作打下基础。论证的主要内容有技术可行性、经费可行性、进度可行性。特别注意对控制项目，要充分重视其可测性和可控性。如果论证结果可行，接着就应该做好签订合同前的准备工作；如果不可行，则应与甲方进一步协商任务委托书的有关内容或对条款进行修改。若不能修改，则合同不能签订。

(6) 甲乙双方签订项目合同书。合同书是甲乙双方达成一致意见的结果，也是以后双方合作的唯一依据和凭证。合同书中应包含如下内容：经过双方修改和认可的甲方"任务委托书"的全部内容，双方的任务划分和各自承担的责任，合作方式，付款方式，进度和计划安排，验收方式及条件，成果归属以及出现违约的解决办法。

伴随着社会主义市场经济的发展，现在的计算机控制工程的设计和实施项目也与其他工程项目类似，越来越多地引入规范的工程招标形式。即先由需求的甲方将所需解决的技术问题和项目要求（主要包括各项技术指标、技术服务内容以及工期等）提出，并写好标书公开向社会招标，感兴趣的单位都可以写出招标书并在规定的时间内投标，开标时间到后，通过专家组开标、评标，确定中标的单位就是乙方。

2. 工程设计阶段

工程设计阶段主要包括组建项目研制小组、系统总体方案的设计、方案的论证和评审、硬件和软件的细化设计、硬件和软件的调试、系统组装。

(1) 组建项目研制小组。在签订了项目的合同或协议后,系统的研制进入设计阶段。为了完成系统设计,应该首先把项目组成员确定下来。项目组应由懂得计算机硬件、软件和有相关控制经验的技术人员组成,还要明确分工并具有良好的协调合作关系。

(2) 系统总体方案的设计。系统总体方案的设计包括硬件总体方案和软件总体方案,比如系统结构、组成方式、硬件和软件功能划分、控制策略和控制算法等,这两部分的设计是相互联系的。因此,在设计时要经过多次的协调和反复,最后才能形成合理的总体设计方案。总体方案要形成硬件和软件的方块图,并建立说明文档,包括控制策略和控制算法的确定等。

(3) 方案的论证和评审。方案的论证和评审是对系统设计方案的把关和最终裁定。评审后确定的方案是进行具体设计和工程实施的依据,因此应邀请有关专家、主管领导及甲方代表参加。评审后应根据评审结果重新修改总体方案,评审过的方案设计应该作为正式文件存档,原则上不应再作大的改动。

(4) 硬件和软件的细化设计。此步骤只能在总体方案评审后进行,如果进行得太早会造成资源的浪费和返工。所谓细化设计就是将方块图中的方块划到最底层,然后进行底层块内的结构细化设计。对硬件设计来说,就是选购模板以及设计制作专用模板;对软件设计来说,就是将一个个功能模块编成一条条程序来实现。

(5) 硬件和软件的调试。实际上,硬件、软件的设计中都需要边设计边调试边修改,往往要经过几个反复过程才能完成,硬件调试主要包括比如元器件测试、电路板调试、子功能模块调试等,软件调试中可以采取设置断点、单步追踪等手段来检验软件模块功能的实现性以及正确性。

(6) 系统组装。硬件细化设计和软件细化设计后,分别进行调试,调试通过后就可以进行系统的整体组装,组装是离线仿真和调试阶段的前提和必要条件。

3. 离线仿真和调试阶段

离线仿真和调试是指主要在实验室而不是在工业现场进行的仿真和调试。离线仿真和调试实验之后,还要进行拷机运行。拷机的目的是要在连续不停机的运行环境中暴露问题和解决问题。

4. 在线调试和投运阶段

系统离线仿真和调试后便可进行在线的调试和运行。所谓在线调试和运行就是将系统和实际生产过程连接在一起,进行现场调试和运行。不管在离线仿真和调试中工作多么认真、仔细,现场调试和运行仍可能出现问题,因此必须认真分析加以解决。系统正常运行后,应当再仔细试运行一段时间,如果不出现其他问题,即可组织验收。验收是系统项目最终完成的标志,应由甲方主持乙方参加,双方协同办理。验收完毕应形成相应的验收文件存档。

10.2　系统的工程设计与实施

实施一个计算机控制系统工程项目的过程中，应安排好科学合理的设计步骤，并严格按照设计的步骤来进行。如果步骤不清晰，或者每一步做什么不明确，就有可能引起实施过程中的混乱甚至返工。10.1节介绍了计算机控制系统工程项目的设计步骤，实际系统工程项目的设计与实现应该按此步骤进行。本节就系统工程设计与实施的具体问题作进一步的讨论，这些具体的问题在实际的设计工作中有着重要的指导意义。

10.2.1　系统总体设计方案

计算机控制系统的总体设计方案是进行系统设计非常关键而重要的一步。因为总体方案的好坏直接关系到整个项目的成本、性能、实施细则等。总体方案的设计要注重对实际问题的调查研究，通过对被控的生产过程、被控对象的生产工艺和流程等深入地了解，才能确定系统的控制任务和目标，提出一个切实可行的总体设计方案与实施方案。

1. 生产现场工艺调研

系统总体方案设计的第一步是现场环境调研和生产工艺的深入调研，通过调研，明确系统要完成的具体任务，然后按照一定的规范、标准和格式，对相应的控制任务和过程进行描述，形成设计任务书，作为整个控制系统设计的依据。

调研的主要任务是弄清控制系统的规模、熟悉工艺流程、明确控制任务、统计输入/输出和控制回路的数量和类型、进行市场咨询或估算投资，最后形成调研报告，并在调研报告的基础上写出可行性的研究方案报告，最后在方案进行多次论证的基础上确定最后的总体设计方案。

2. 硬件总体设计方案

根据合同的设计要求和已经作过的初步方案，开展系统的硬件总体设计。硬件总体方案设计主要包含以下几个方面的内容。

(1) 确定系统的结构和类型。根据系统要求，确定采用开环还是闭环控制。如果是闭环控制还需进一步确定是单闭环还是多闭环控制。在实际中可供选择的控制系统类型有：操作指导控制系统、直接数字控制(Direct Digital Control，DDC)系统、监督计算机控制(Supervisory Computer Control，SCC)系统、分散型控制系统(Distributed Control System，DCS)、工业测控网络系统等。

(2) 确定系统的构成方式。确定系统的构成方式主要是选择机型，在计算机控制系统中，目前常用的产品有可编程序控制器、工控机、单片机、DSP、可编程智能调节器等。一般应优先选择总线式的工业控制机来构成系统。工控机具有系列化、模块化、标准化和开放式的系统结构，有利于系统设计者在系统设计时能根据要求任意地选择，像搭积木般地组建控制系统。这种方式可以提高研制和开发速度，提高系统的技术水平和性能，同时增加可靠性。

当然,如果控制系统的规模较大、自动化程度较高时,可以选用现场总线控制、PLC 等组成控制网络;如果是小型控制系统或者智能型仪器仪表就能完成的,可以采用单片机系列。在实际工程中选择何种控制器,应根据具体的控制规模、工艺要求、控制特点和所完成的工作来确定。

(3) 现场设备选择。工业控制现场设备主要包括传感器、变送器和执行机构。这些现场装置的选择要正确合理,它们是影响系统控制精度的重要因素之一。

(4) 其他方面的考虑。总体方案设计中还要考虑人机联系的方式、系统机柜或者机箱的结构设计以及系统抗干扰等方面的问题。

3. 软件总体设计方案

依据用户任务的技术要求和已作过的初步方案,进行软件的总体设计。软件总体设计的内容主要是确定软件平台、软件结构,分解任务,确定系统的数学模型、控制策略和控制算法等。软件总体设计和硬件总体设计一样,也是采用结构化的"黑箱"设计法。自上而下或者自下而上地画出软件结构的方框图,然后再将大的方框分解成小的方框,逐级细化,直到能表达清楚功能为止。

4. 系统总体方案

硬件设计和软件设计是密不可分的,将它们组合在一起便构成了整个系统的总体方案。总体方案是项目具体实施时的依据,因此应在工艺技术人员的配合下,从可行性、合理性以及经济性等方面进行反复论证,仔细地斟酌。总体方案论证可行后,要形成文件,建立总体方案文档。系统总体方案应包括以下几个方面的内容。

(1) 系统的主要功能、技术指标、原理性方框图及文字说明。

(2) 控制策略和控制算法。

(3) 系统的硬件结构及配置,主要的软件功能、结构及框图。

(4) 方案比较和选择。

(5) 保证性能指标要求的技术措施。

(6) 抗干扰和可靠性设计。

(7) 机柜或机箱的设计。

(8) 经费和进度计划的安排。

(9) 对现场条件的要求。

总之,系统的总体设计方案反应了整个系统的综合情况,要从合理性、经济性、可靠性以及可行性等角度来评价系统的总体方案。总体方案通过后,便可形成作为系统设计依据的系统总体方案图和设计任务书,从而能针对性地指导系统设计的具体过程。

10.2.2 硬件设计

在系统总体方案确定以后,就要详细进行硬件的细化设计。对于不同的系统结构和类型有不同的设计与实现方法。采用总线式的工业控制计算机进行系统的硬件设计是目前广泛使用的结构类型,可以解决工业控制中的众多问题。由于总线式的工业控制计算机具有高度的模块化和插拔式结构,使得采用组合方式可以大大简化计算机控制系统的设计。采

用总线式的工业控制计算机，当控制功能变换时，只需简单地更换几块模板即可实现。

1. 系统总线的选择

总线就是指计算机各个模块之间互连和传送信息的一组信号线。系统采用总线结构，具有很多优点。采用总线结构，可以简化硬件设计，用户可根据需要直接选用符合总线标准的功能模板，而不必考虑模板插件之间的匹配问题，从而使系统硬件设计大大简化；系统可扩展性好，仅需将按总线标准研制的新的功能模板插在总线槽中即可，方便和其他控制主机的通信；系统更新性好，一旦出现速度更快、功能更强的微处理器、存储器芯片和接口电路，只要将这些新的芯片按总线标准研制成各类插件，即可取代原来的模板而升级更新系统。

(1) 内部总线选择。内部总线是指计算机系统各个组成部分之间进行通信的总线，按其功能可分为数据总线(Data Bus，DB)、地址总线(Address Bus，AB)、控制总线(Control Bus，CB)和电源总线(Power Bus，PB)四种，每种型号的计算机都有自身的内部总线。常用的工业控制计算机的内部总线有两种，即 PC 总线和 STD 总线。设计时可根据需要进行选择，一般常采用 PC 总线进行系统的设计，即选用 PC 总线标准的工业控制计算机。

(2) 外部总线选择。外部总线是指计算机与计算机之间、计算机与其他智能设备之间或者智能外设之间进行通信的总线。它包括并行通信总线(IEEE-488)和串行通信总线(比如 USB、RS-232C、RS-422、RS-485)。对于远距离通信、多站点互连的通信，还可以选用 RS-422 和 RS-485 来组建通信网络。在系统设计中具体选择哪一种，设计人员应根据控制系统的需求、通信距离、速率、系统拓扑结构、通信协议等要求来综合分析确定。但需要说明的是 RS-422 和 RS-485 总线在工业控制机的主机中没有现成的接口装置，必须要选择相应的通信接口电路或通信接口板，这势必会增加系统的成本。

2. 控制主机选择

在总线式的工业控制机中，可供选择的计算机有许多系列和种类，其 CPU、内存、硬盘、主频、显示器等也有多种规格，设计人员应根据要求主要从字长、处理速度等方面来合理地选择。如果控制现场环境比较好，对可靠性的要求又不是特别高，可以选择普通的个人计算机来作为控制机，也可以选用单片机、PLC 等来实现计算机控制功能。在主机的配置选择上，以留有余地、满足需要为原则，不一定要追求选择最高档的配置。

3. 输入输出通道选择

一个典型的计算机控制系统，除了工业控制计算机的主机以外，还必须配备有连接计算机和生产过程之间信息传递和变换的通道，即各种输入输出通道。对于总线式的工业控制计算机，生产厂家通常以模块的形式提供，方便使用。输入输出通道模板主要包括数字量输入/输出(即 DI/DO)、模拟量输入/输出(AI/AO)等模板。

(1) 数字量输入/输出(即 DI/DO)模板。数字量输入/输出模板的种类很多，通常可分为 TTL 电平的 DI/DO 和带光电隔离的 DI/DO。通常与工业控制机共地装置的接口采用 TTL 电平，而其他装置与工业控制机之间则采用光电隔离。对于大容量的 DI/DO 系统，往往选用大容量 TTL 电平的 DI/DO 板，而将光电隔离以及驱动功能配置在工业控制机总线之外的非总线模板上，比如继电器板等。

(2) 模拟量输入/输出(AI/AO)模板。AI/AO板卡包括A/D、D/A板以及信号调理电路等。AI模板可以接收的输入信号是-5～0V、0～5V、1～5V、0～10mA、4～20mA信号以及热电阻、热电偶和各种变送器的信号。AO模板输出的信号可能是0～5V、1～5V、0～10mA、4～20mA等信号。在选择AI/AO模板应根据AI/AO路数、分辨率、转换速度和量程范围等技术指标。

系统中的输入输出模板的类型、数量等应该按照具体的控制系统的输入输出参数的种类、数量及控制要求来确定,并适当地留有余量,以将来扩充需要时使用。

4. 现场设备选择

(1) 变送器和传感器选择。在计算机控制系统中,为了正确地指导生产操作、保证生产安全、产品质量和实现生产过程的自动控制,一项必不可少的工作是准确而及时地检测生产过程中的各个有关参数,例如压力、流量、物位及温度等。这个任务是由检测仪表来完成的。传感器和变送器都属于检测仪表,传感器是将被测物理量参数转换为一定的便于传送的信号(例如电信号或气压信号,通常是0～10mA、4～20mA电流信号等),并且其输出信号与被测变量呈一定对应的关系。当传感器的输出为单元组合仪表中规定的标准信号时,通常又称为变送器。变送器或传感器的输出信号经输入通道被送至工业控制机中进行处理,实现数据的采集。

常用的变送器有DDZ-II型、DDZ-III型以及新发展起来的DDZ-S型,早期的以电子管为主要器件的DDZ-I型基本淘汰。其中DDZ-II型输出的是直流0～10mA信号,采用四线制的220V供电;而DDZ-III型输出的是直流4～20mA信号,供电电源为直流24V且采用二线制,DDZ-III比DDZ-II型的变送器性能好,使用更方便。DDZ-S系列变送器是在总结DDZ-II和DDZ-III变送器的基础上,广泛吸收国外同类变送器的先进技术,采用模拟技术与数字技术相结合而开发出的第4代变送器。近几年来,还出现了以微处理器为基础的智能变送器,以及现场总线仪表的推广和使用,为工程设计者提供了更大的选择空间。

常用的变送器有压力变送器、温度变送器、液位变送器、流量变送器以及各种电量变送器等。系统设计人员可根据被测参数的种类、量程、被测对象的介质类型和环境等来选择变送器的具体型号。

(2) 执行机构选择。执行机构一般接收来自控制器的控制信号,将其转换成相应的角位移或直线位移去操纵调节机构改变控制量,从而使被控变量符合系统预期要求,是控制系统中必不可少的重要组成部分。对于计算机控制系统来说,它的作用是接收计算机发出的控制信号,并把它转换成执行机构的动作,使生产过程按照预先规定的要求正常进行。

执行机构按使用的能源分为电动、液动和气动三种。气动执行器是以压缩空气为动力能源的一种自动执行器。气动执行机构结构简单、价格低,输出推力大,防火防爆,动作可靠,维修方便,适用于防火、防爆场合,因此被广泛应用在化工、炼油生产中,在冶金、电力、纺织等工业部门也得到大量使用。气动执行机构与计算机的连接极为方便,只要将电量信号经电气转换器转换成标准的气压信号之后,即可与气动执行机构配套使用,但是需要额外配置气源以及电气转换装置。

电动执行器是以电为能源的执行机构。电动执行机构是工程上应用最多、使用最方便的一种执行器,它的主要特点是能源取用方便,信号传输速度快,传送距离远;便于集中控

制；停电时执行器保持原位不动，不影响主设备的安全；灵敏度和精度较高；与电动调节仪表配合方便，安装接线简单。缺点是结构复杂、推力小、价格贵，平均故障率高于气动执行器，适用于防爆要求不高及缺乏气源和使用数量不太多的场合。

液动执行器的特点是推力大、精度高，但在实际中应用较少。

在计算机控制系统中，电动执行机构可以直接接收来自工业控制机的 0～10mA 或者 4～20mA 的输出信号，实现对被控对象的控制。而对于选用的气动执行机构，在使用时要先将 0～10mA 或者 4～20mA 的电信号经过电气转换器转换成标准的 0.02～0.1MPa 的气压力信号，才可与气动执行机构配套使用。

常用的执行器有电动机、变频器、调节阀、电磁阀、继电器等，对于只实现开关控制的场合，还可以选用各种有触点和无触点的开关，作为执行机构来实现开关动作。

在系统工程设计中，要根据系统的具体要求来选择执行机构，比如要实现连续精确的控制，必须选用气动或者电动调节阀，而对于控制要求不高的可选用电磁阀。

5. 操作控制面板设计

操作控制面板也就是操作控制台，是人机对话的重要纽带，也是计算机控制系统中的重要设备之一。根据具体的系统情况，操作面板可大可小，大到一个庞大的操作台，小到几个功能键按钮和开关，一般需根据实际情况自行设计。在设计中应遵循操作方便、使用简单、安全可靠、布局美观、符合人体工程学的原则。

操作面板完成的主要功能有：通过面板操作来监视程序执行情况，打印、显示、存储中间结果或最终结果，选择工作方式以及控制回路，完成手动/自动的无扰动切换，完成各种监控画面的显示等。为了完成这些功能，操作台上一般都有操作器件、显示器件、打印和报警装置等。

操作器件主要是键盘按键，还可以有各种按钮和开关，开关的形式可以采用拨动开关、旋转开关和滑动开关等。显示器件主要有 LED 显示器、LCD 显示器、CRT 显示器和指示灯等。打印装置主要是打印机。报警装置一般是警铃或扬声器等。

10.2.3　软件设计

由于工业控制计算机中大都配有实时操作系统、实时监控程序、各种控制、运算软件模块以及组态软件等，可以使设计者在最短的时间内，根据控制要求，选择所需的模块进行组态，再开发出控制目标系统软件，所以用工业控制机来组建计算机控制系统不仅能大大减少硬件设计的工作量，而且还能减少系统软件设计的工作量。因此根据项目资金情况应首先选用质量和信誉好的品牌工业控制机，这样在充分保证硬件质量的同时，也能够得到较为丰富的软件支撑。

一般的工业控制机把工业控制所需的各种功能以模块形式提供给用户。比如算法策略模块(多为 PID)、运算模块、输入输出模块、打印模块等，系统设计者根据控制任务要求，选择所需的模块就能生成系统控制软件。但是并不是所有的工业控制机都能给系统设计者带来方便，有些工控机只能提供硬件设计的方便，而应用软件需要自行开发。比较常见的是需要设计者自行开发控制策略与算法等。自行开发控制软件时，应先画出系统程序总体流程图和各个功能模块图，再选择实现的程序设计语言，然后就可以编制实现程序了。具体的程

序设计要考虑以下几个内容。

1. 程序结构规划

在系统总体方案设计中，系统的各个模块之间存在着各种因果关系，模块之间相互要进行各种信息传递。比如数据处理模块和数据采集模块之间的关系，数据采集模块的输出信息就是数据处理模块的输入信息，同样，数据处理模块和显示模块、打印模块之间也有这种产销关系，即数据处理模块的输出可能又是控制模块、显示模块和打印模块等的输入。各模块之间的关系体现在程序的流程上和接口条件上，即输入条件和输出结果上。为了防止接口条件出错，将每一个执行模块要用到的参数、要输出的结果列出来，对于与不同模块都有关的参数，只取一个名称，以保证同一个参数只有一种格式，并为每一个参数规划一个数据结构和数据类型。

数据类型分为逻辑型和数值型两种，但通常将逻辑型数据归到软件标志中去考虑，用于设置软件标志。数值型又可分为定点数和浮点数。定点数有直观、运算速度块、编程简单的优点，其缺点是可以表示的数值范围小，运算中容易溢出。浮点数则相反，可表示数值的动态范围大、精度高、不易溢出，但是编程比较复杂，运算速度低。

如果某参数为一系列有序数据的集合，比如采样信号序列，则考虑的不只有数据类型的问题，还存在数据的存储格式，也就是数据结构问题。

2. 资源分配

在完成数据类型和数据结构的规划后，便可开始分配系统的资源了。软件设计时应了解的系统资源包括 RAM、ROM、定时器/计数器、中断源、I/O 地址等。ROM 资源用来存放程序和表格。定时器/计数器、中断源、I/O 地址在任务分析时已经分配好了。因此，资源分配的主要工作是 RAM 资源的分配。RAM 资源规划好后，应列出一张 RAM 资源的详细分配清单，作为编程依据。

3. 实时控制软件设计

实时控制软件是软件设计中的主要部分，主要包括以下几个方面的设计。

(1) 数据采集和数据处理程序。数据采集程序主要包括多路信号的采样、输入变换和存储等。数据处理程序包括各种数字滤波、线性化处理和非线性补偿、标度变化和越限报警程序等。

(2) 控制算法程序。控制算法主要是实现某种控制规律的计算并产生控制量。其中包括数字 PID 控制算法、最少拍控制算法、串级控制、前馈控制、预测控制、模糊控制、解耦控制和最优控制等。实际设计时可以选择一种或几种合适的算法来实现控制任务，算法的实现可以使用这些控制算法现有的软件模块，也可以自己编制来实现。

(3) 控制量输出程序。控制量输出程序主要完成对控制量的处理，比如上下限和变化率处理、控制量的变换及输出，并驱动执行机构和电气开关动作。控制量输出通常包括模拟量和开关量输出两种。其中模拟量主要有 D/A 转换板来实现输出，通常为标准的 0～10mA 或者 4～20mA 直流信号，它们可以驱动各种执行机构动作。开关量主要有 DO 板来实现输出，驱动各种电气开关。

(4) 实时时钟和中断处理程序。实时时钟是计算机控制系统一切与时间有关过程的运行基础,它包括绝对时钟和相对时钟。绝对时钟与当地的时间同步,有年、月、日、时、分、秒等功能。相对时钟与当地时间无关,只与时间起点有关,一般只要时、分、秒就可以,在某些场合可能需要精确到更小的时间单位。

计算机控制系统中许多任务如数据的采集、实时显示打印等都必须利用实时时钟来实现。实时时钟可通过软件定时中断或硬件电路来实现。

另外,对于事故报警、掉电检查及处理、重要事件的处理等功能的实现常常使用中断技术,以便计算机能对事件做出及时处理。事件处理常用中断服务程序和相应的硬件电路来完成。

(5) 数据管理程序。数据管理程序主要用于生产的管理和监督,比如画面运行显示、数据记录、变化趋势显示、报警记录和统计报表等。

(6) 数据通信程序。随着现代工业发展的不断深入,数据通信越来越多地用于计算机控制系统中。数据通信主要完成计算机与计算机之间、计算机与智能设备之间的实时信息传递和交换。比如在分散控制系统、现场总线系统、工业控制网络和企业综合自动化系统中都得到了广泛的应用。

(7) 人机界面程序。人机界面程序主要是指面板操作管理程序,包括键盘、开关、拨码盘等信息输入处理程序,显示器、指示灯、监视器和打印机等输出程序等。

(8) 公共服务程序。公共服务程序主要包括函数运算程序、基准运算程序以及数码转换程序等。

10.2.4　系统调试与运行

控制系统的调试与运行分为离线仿真与调试阶段和在线调试与运行阶段。离线仿真与调试阶段一般是在实验室或非工业现场进行,在线调试与运行是在生产工程的工业现场进行。其中离线仿真与调试阶段是基础,主要是检查硬件和软件的整体性能,为现场投入运行做准备,在线调试和现场投入运行是对整个系统的实际考验与检查。系统调试的内容很丰富,可能碰到的问题也千变万化,解决的方法也是多种多样的,没有一个统一的模式。

1. 离线仿真和调试

(1) 硬件调试。对于硬件调试通常是按模块进行的,所有的模块单独调好后,再进行硬件联调。这样做可使问题局限在一个模块内,以便于及时发现和解决。

对于各种标准功能模板,按照说明书检查其主要功能;对于 A/D、D/A 模板的调试,必须准备好信号源、电压表和电流表等,对这两类模板应先检查信号的零点和满量程,然后再分档检查;对于 DI/DO 模板,可利用开关量输入输出程序来检查,在输入端加开关量信号,检查读入状态的正确性,并可在输出端用万用表检查输出状态的正确性;同时对于现场的检测仪表和执行机构,在使用安装前必须按说明书要求进行校验;对于计算机控制系统网络,还要调试其通信功能和数据传输正确性的验证。

(2) 软件调试。软件调试包括软件模块调试和软件联调,其顺序一般是对各个子程序、功能模块、主程序分别调试以及整体程序的联合调试。软件模块调试一般是借助与目标系统同机种的高中档微机系统或专用微机开发系统作为工具,独立于硬件进行。软件联调既

可以在与实际环境尽可能相近的开发机提供的环境中进行，也可以在硬件联调全部通过的基础上进行。总的原则是尽量把可能保留的大部分问题局限在软件的范围以内，以避免软、硬件中的问题相互交叉，从而减轻调试的难度和工作量。

软件调试的方法一般是自下而上逐级调试。对于比较简单的程序，利用开发装置或仿真器以及计算机提供的调试程序就可以进行调试。程序设计一般采用汇编语言和高级语言来编程。对于实时性和处理速度要求较高的部分采用汇编语言来实现，比如数据采集、中断、控制输出等；对于实时性和速度要求不高的部分用高级语言来实现，比如图形显示、打印以及统计报表等。

(3) 系统仿真。硬件和软件的单独调试通过后，并不意味着系统的设计和离线调试已经结束，在现场总装前还应在实验室再进行一次系统软、硬件的仿真联调，即通过模拟被控对象、控制系统工作的实际环境等，研究、分析所设计系统的性能。

在实验室进行的系统仿真试验，应尽量采用全物理仿真或半物理仿真。工作状态或试验条件越接近真实情况，仿真的效果就越好。对于纯数据采集系统，一般可以做到全物理仿真，但是对于控制系统，不可能将实际生产过程搬到实验室中来进行，做到全物理仿真几乎是不可能的，只能尽量模拟实际装置来做功能试验，同时还必须根据实际运行环境的要求，进行特殊运行条件的考验。比如高温和低温剧变运行试验，可靠性与抗干扰性试验，电源电压突变和掉电保护试验等。

通过系统仿真试验，可以评价所设计的控制系统性能优劣，发现硬件和软件缺陷并及时给予修改。

2. 在线调试和运行

尽管离线仿真和调试工作很仔细，检查也很严格，但是仍然需要接受实践的考验。在现场进行调试和运行的过程中，工程实践人员要与用户密切配合，在实际运行前制订一系列调试计划、实施方案、安全措施、分工合作细则等。现场调试与运行过程应遵循从小到大，从易到难，从手动到自动，从简单回路到复杂回路逐步过渡的原则。为了做到稳妥地实现计算机控制，现场安装及在线调试前先要进行下列检查。

(1) 检测元件、变送器、显示仪表、调节阀等必须通过校验，保证精确度要求。作为检查，可进行一些现场校验。

(2) 各种电气接线和测量导管必须经过检查，保证连接正确。例如，孔板的上下游接压导管要与差压变送器的正负压输入端极性一致；热电偶的正负端与相应的补偿导线相连接，并与温度变送器的正负输入端极性一致等。引压导管和气动导管必须畅通，不能中间堵塞。

(3) 对在流量中采用隔离液的系统，要在清洗好引压导管以后，灌入隔离液(封液)。

(4) 检查调节阀能否正确工作。旁路阀及上下游截断阀关闭或打开，要搞正确。

(5) 检查系统的干扰情况和接地情况，如果不符合要求，应采取措施。

(6) 对安全防护措施也要检查。

经过检查并已安装正确后，即可进行系统的投运和参数的整定。投运时应先切入手动，等系统运行接近于给定值时再切入自动。

计算机控制系统的投运是一个系统工程，是对计算机控制系统设计性能的全面检查和

考核，要特别注意到一些容易忽视的问题，如现场仪表与执行机构的安装位置，现场校验，各种接线与导管的正确连接，系统的抗干扰措施，供电与接地，安全防护措施等。在现场调试的过程中，往往会出现错综复杂、时隐时现的奇怪现象，一时难以找到问题的根源，此时此刻，计算机控制系统设计者们要认真地共同协作分析，以便尽快地找到问题的根源所在。

一般在计算机控制系统运行正常并且试运行一段时间后，就可以邀请相关人员进行组织验收。一般验收是由甲方主持、乙方参加，双方协同进行的。验收完毕后还要形成相应的存档文件进行存档。

10.3　基于 OPC 和 Simulink 的实时过程控制系统设计

OPC 是用于工业控制领域的一个技术规范和工业标准，它是由世界上一些著名的自动化系统和硬件、软件公司及 Microsoft（微软）紧密合作而建立的。OPC 是专为在现场设备、自控应用、企业管理应用软件之间实现系统无缝集成而设计的接口规范。

MATLAB 和 Simulink 在控制领域的应用已经非常广泛，MATLAB 具有强大的工程计算能力，易于实现复杂的控制算法，而 Simulink 有着极强的仿真能力和数据显示能力。但由于 MATLAB/Simulink 只停留在纯数字仿真上，无法验证算法的有效性，大量的先进算法都只是在理论和纯数字仿真的基础上实现。如果能通过 Simulink 直接实时控制现场设备，则可以使工程设计及研究人员直接观测控制效果，易于进行控制算法的设计和控制效果的分析，提高研究工作效率。

本设计的实时过程控制系统是以 OPC 技术为基础，建立 Simulink 与计算机控制系统控制器之一 PLC 的结合。在控制平台的建立过程中，MATLAB 与 WinCC 的通信是首先要解决的问题。OPC 技术是 Windows 应用程序间数据交换的主流技术，在最新版本的 MATLAB 中添加了工具箱 OPC Toolbox，利用该工具箱可以实现在 MATLAB 中获取外部实时数据。本控制系统所使用的 MATLAB 版本为 R2008a（即 MATLAB 7.6）。

10.3.1　控制系统的结构

Simulink 与 PLC 的实时过程控制系统结构如图 10.1 所示。系统以过程控制装置为控制对象，西门子 S7-300 PLC 作为下位机对现场设备数据进行采集，以组态软件 WinCC 为数据总控平台，并作为 OPC 服务器。以运行 MATLAB/Simulink 的电脑作为 OPC 客户端，进行算法设计、实时控制和控制结果分析等，WinCC 与 S7-300 PLC 通过 MPI 通信。OPC 服务器和 OPC 客户端以 IT 网络互连。OPC Server 和 OPC Client 都要运行 dcomcnfg 进行 DCOM 设置，这是通过 OPC 技术实现数据交换的基础。

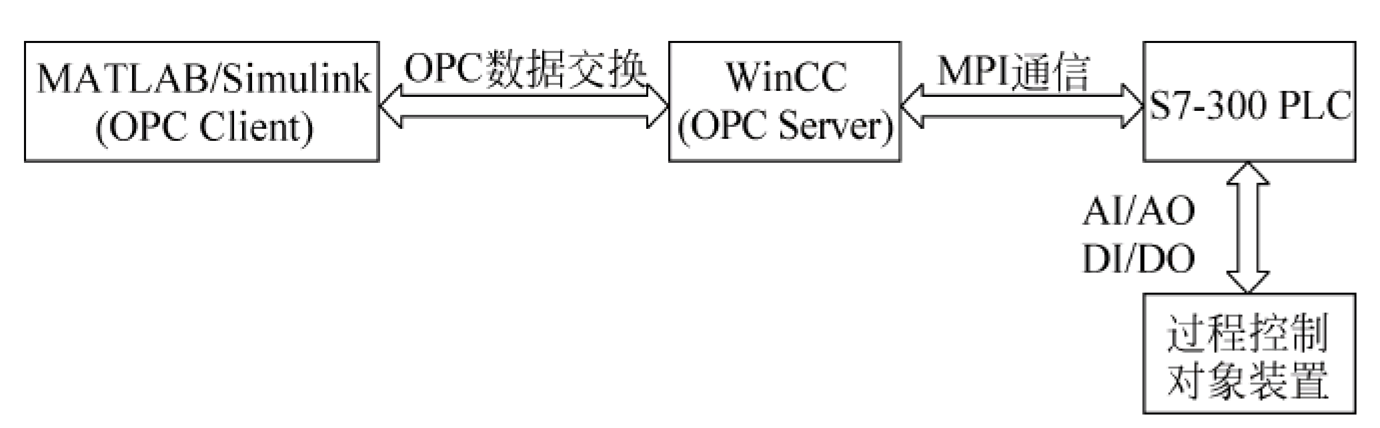


图 10.1　MATLAB/Simulink 在线实时过程控制系统结构图

10.3.2 控制系统原理

本设计以过程控制单容水箱液位的闭环控制为例。假设液位设定值为 300mm,控制器为 MATLAB 及 PLC,液位变送器作为检测装置,调节阀作为执行机构来构成一个单回路的恒液位控制系统。

MATLAB/Simulink 主要实现 PID 运算,PLC 主要实现模/数、数/模转换及信号处理。液位变送器将标准的电信号(4～20mA)输入给模拟量输入模块(SM331),经 A/D 转换后进行标度转换,变为工程液位值,WinCC 从 PLC 中读取实时液位值,通过 OPC 接口技术传输到 MATLAB 的 Simulink 中。在 Simulink 中与液位设定值(阶跃输入为 300)比较,并进行 PID 运算,将运算结果再通过 OPC 技术传送给 WinCC,并经 WinCC 写入 PLC 中,最后经过标度反变换后,输出到模拟量输出模块(SM332),经 D/A 转换后变为电流信号送给智能调节阀,用来控制电动调节阀的开度,通过它控制流入水箱的流量,实现对液位的闭环控制。

10.3.3 PLC 程序设计

在 S7-300 PLC 程序的组织块 OB1 中进行启动和停止的控制。而液位的采集、控制输出、数据转换存放在中断服务程序 OB35 中。OB35 每 100ms 定时中断一次,即每 100ms 采集一次液位信号并将 Simulink 中的 PID 运算结果输出给电动调节阀。PLC 程序流程图如图 10.2 所示。

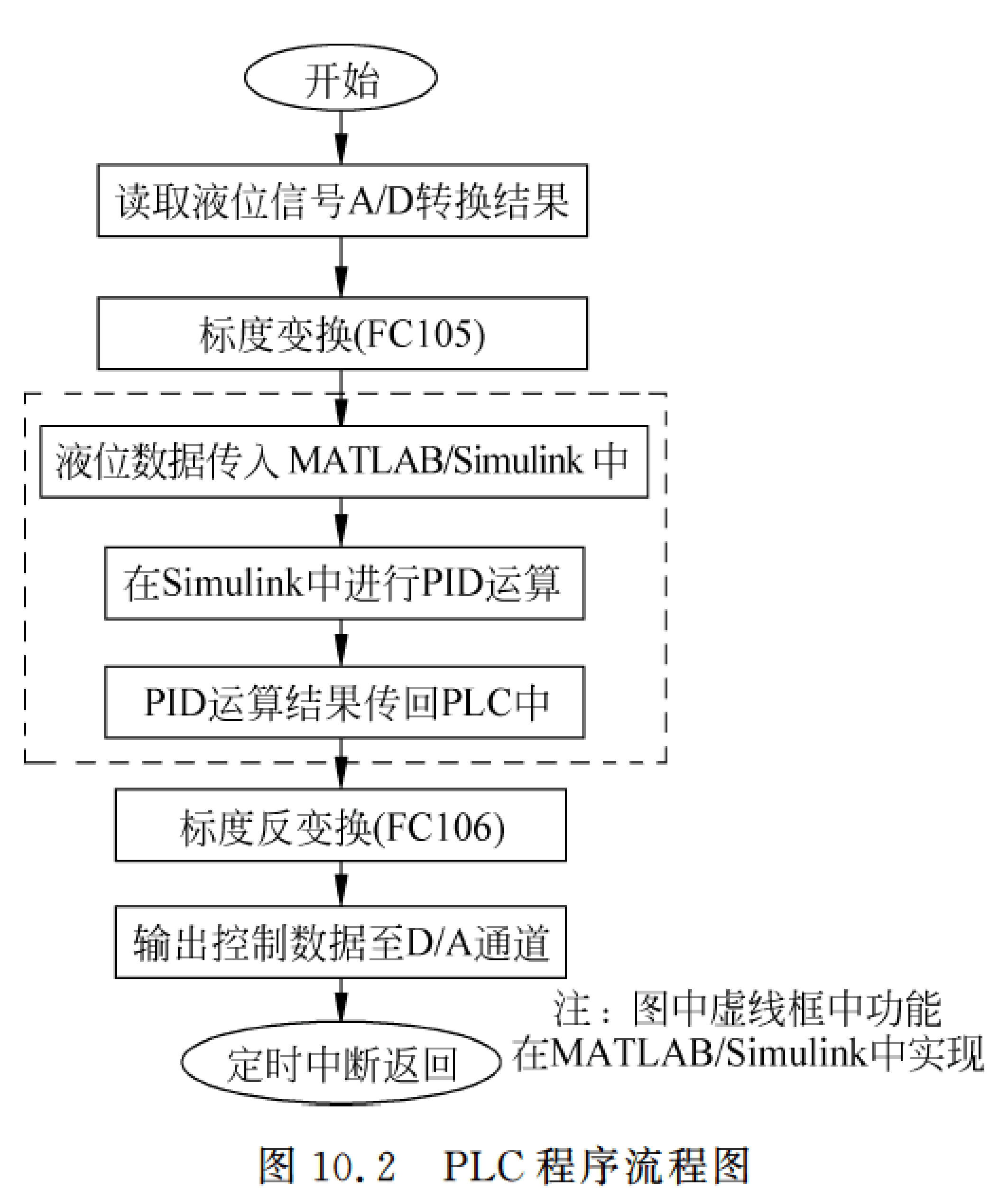


图 10.2 PLC 程序流程图

10.3.4 MATLAB/Simulink 在线连接

WinCC 作为 OPC 服务器,MATLAB 作为 OPC 客户端,获取现场过程数据并写入相关的控制算法中。其通信流程如图 10.3 所示。

在 MATLAB 的 OPC Tool 界面中,先把主机(系统中用 Localhost)添加到 OPC 页面中。注意此时一定要让 WinCC 服务器处在“运行(激活)”状态,否则会出现“服务器没有注册类别”或者“找不到服务器”的错误提示。

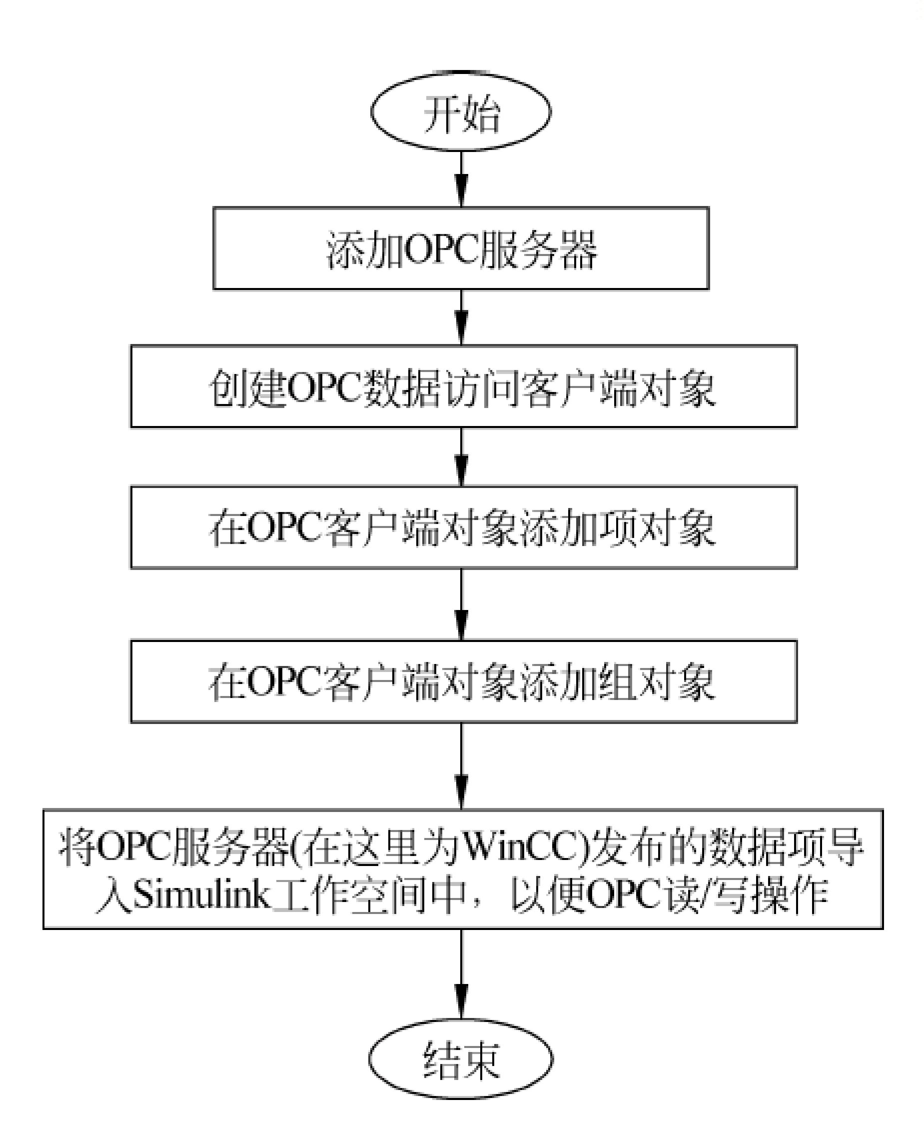


图 10.3　Simulink 和 WinCC 通信连接流程图

在 Localhost 下拉菜单中会出现希望联通的 OPC 服务器 OPCServer. WinCC，选择 WinCC 服务器，并右击选择 Create Client 来创建和 WinCC 服务器联通的 OPC 客户端。

在已创建的 Client 中选择 Add group 来添加工作组，在这里添加进行水箱液位控制的工作组 yeiwei，然后单击 Connect 创建和 OPC 服务器的连接。

在已创建的工作组 yeiwei 中选择 Add Item，在 Add Item 对话框中选择要添加的 WinCC 中的变量 yeweil 和 yeweixie。然后单击 Add 即把这两个变量联通到 OPC Client 中来，如图 10.4 所示。

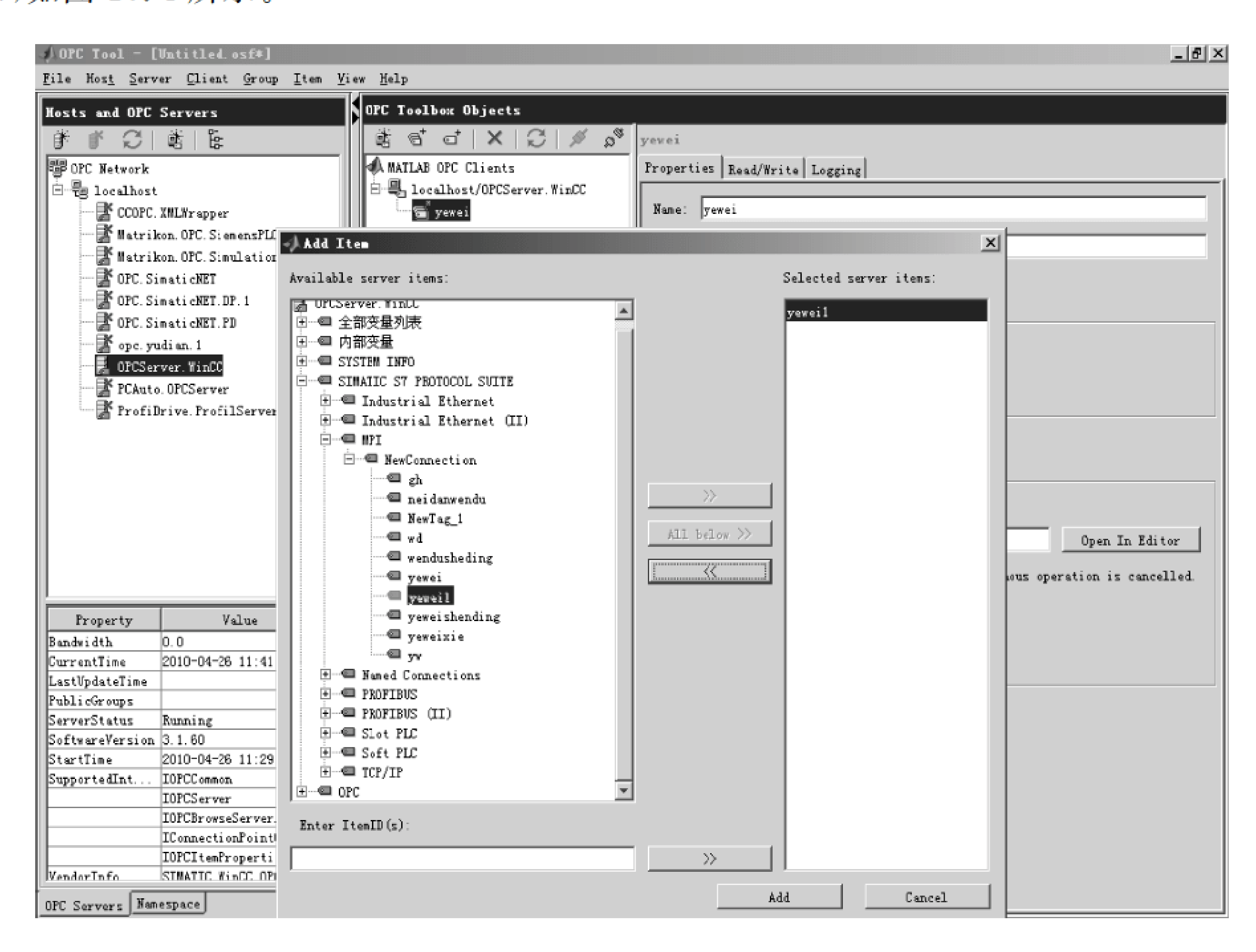


图 10.4　添加 OPC 工作组及变量

变量 yewei1 即 OPC Read 操作对应的变量，是本控制项目中水箱液位的采集值，它和 PLC 中的物理地址 MD80 连接。变量 yeweixie 即 OPC Write 操作对应的变量，表示 Simulink 中 PID 控制器运算的输出值，最终写入到 PLC 物理地址的 MD90 中，用于对调节阀开度进行控制。

OPC 写的过程一定要与实际的 PLC 相连接并且使 PLC 处于运行状态，否则 OPC 的连接状态会显示：Bad。

10.3.5 建立 Simulink 实时控制工作空间

在 OPC Clients 界面中，右击项目组 yewei 选择 Export To 中的 Simulink OPC Read/Write，把所需变量即当前液位反馈信号 yewei1 及 PID 运算结果输出信号 yeweixie 添加到 Simulink 仿真环境下，如图 10.5 所示。随后在 Simulink 仿真窗口中就会出现 OPC Read/Write 的功能块。

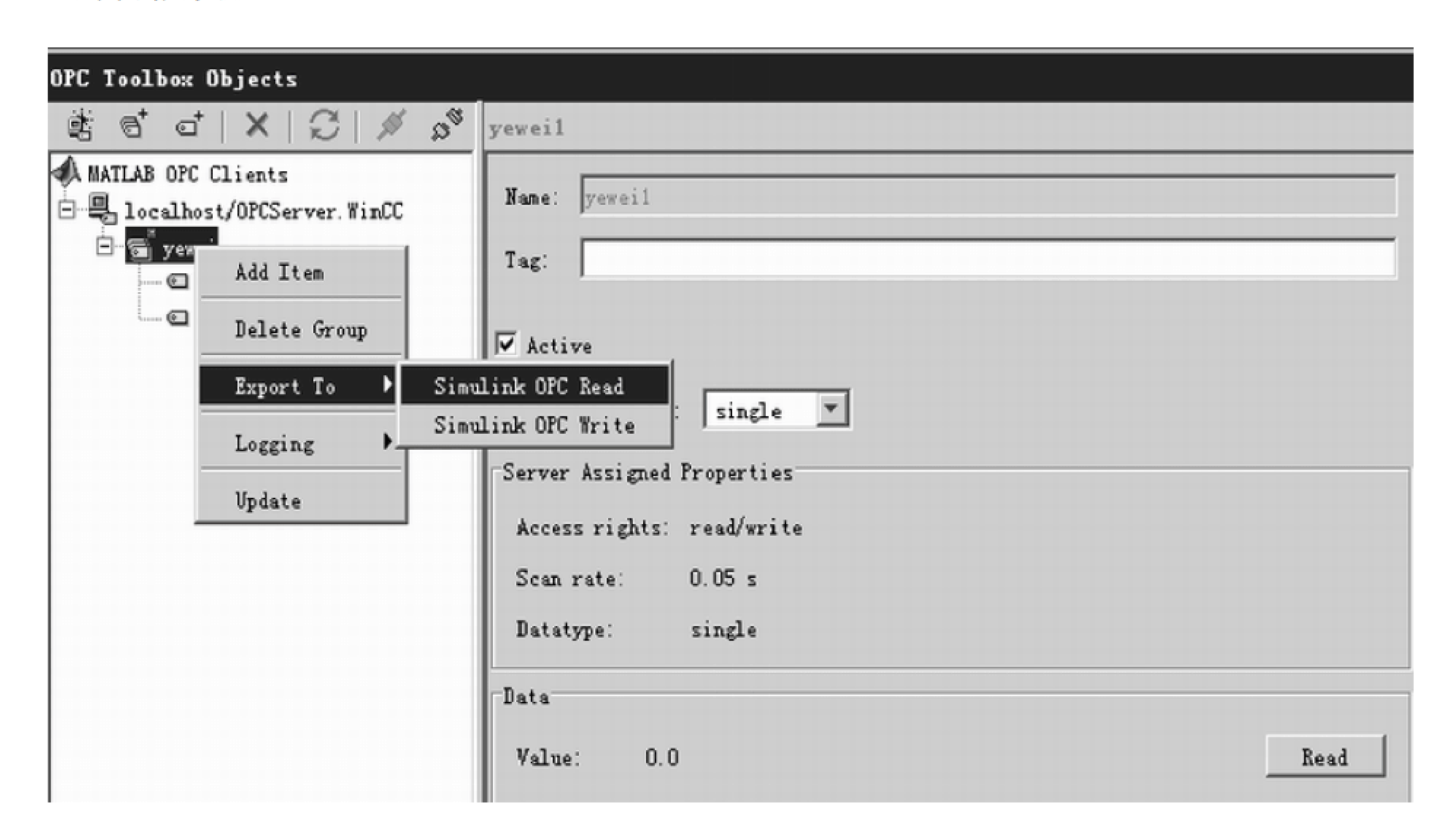


图 10.5 把变量导入 Simulink 中

在 Simulink 仿真窗口中双击 OPC Read/Write，进行对象的属性设置和采样时间设置，本系统设计中采样时间均设为 0.01s。这个时间设置和 PLC 中 OB35 的定时中断时间设置 100ms 是一致的。有了当前液位反馈信号和控制信号输出通道，即可按需要搭建各种控制算法。这里搭建了一个 PID 控制算法结构，如图 10.6 所示。

在 Simulink 中建立工作空间，Simulink 仿真参数需要进行设置，仿真结束时间（stop time）设为 inf（为了使系统运行在无时间限制条件下而设置），Simulink 仿真算法设置为 Fix-step 固定步长类型，并选择 ODE1（EULER）算法。设定值为 300，经过调试，得到了一组效果比较理想的 PID 参数：Kp=100，Ti=0.1，Td=1。

控制系统的响应曲线如图 10.7 所示，液位虽然有一定的超调，但是很快就能返回到设定值附近，最终基本稳定到设定值，说明控制效果比较理想。

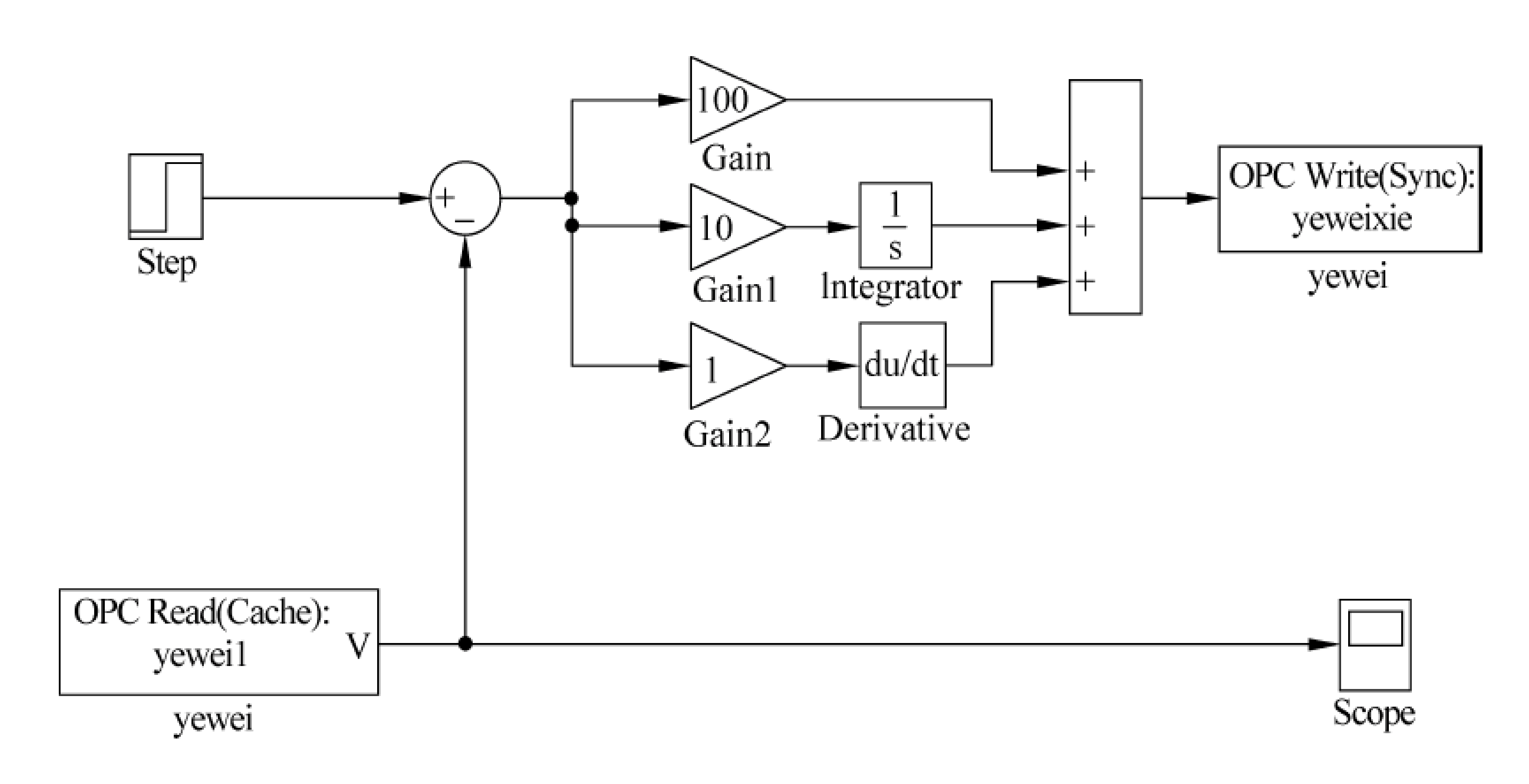


图 10.6　在 Simulink 工作空间中搭建 PID 控制算法

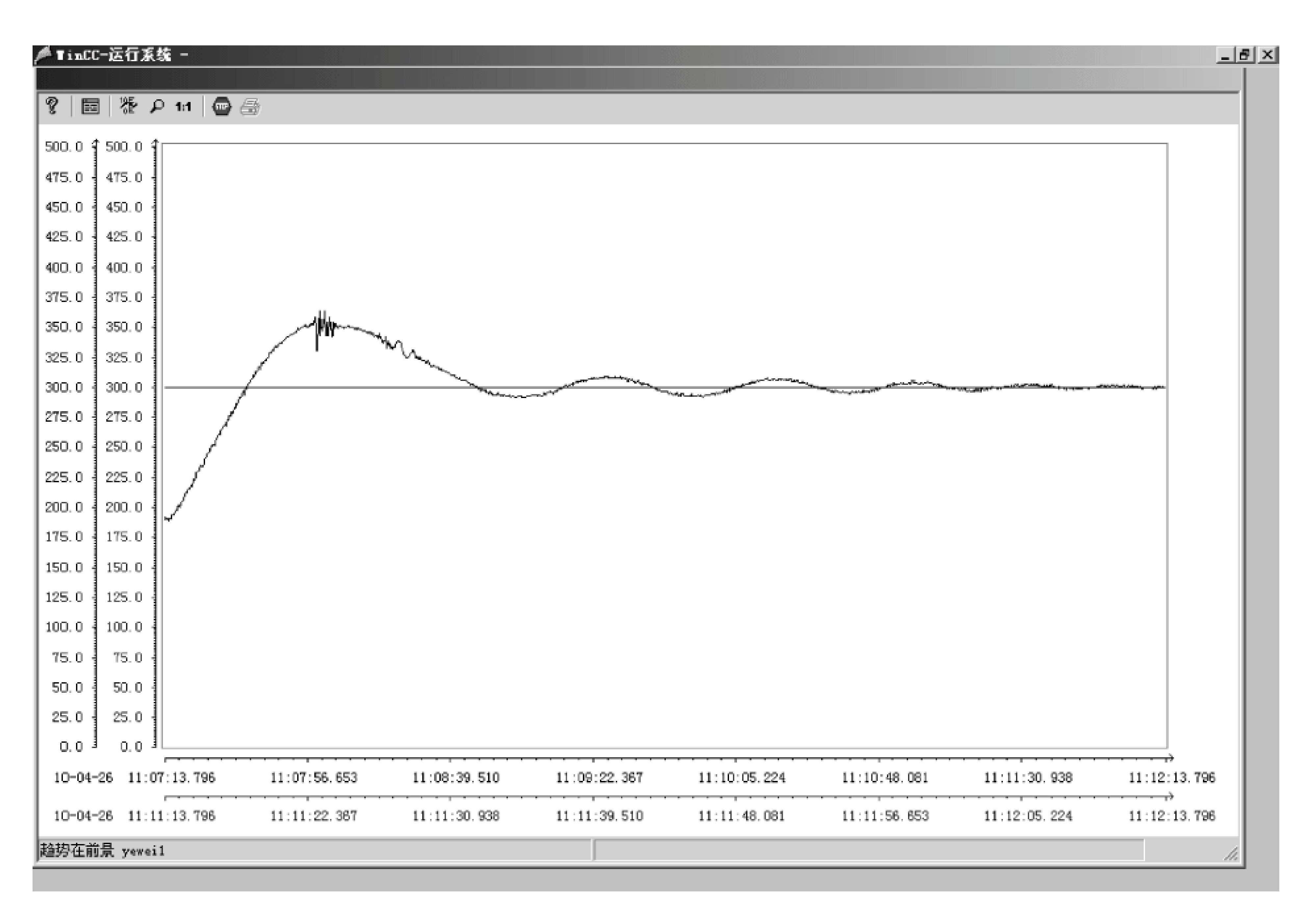


图 10.7　水箱液位控制响应曲线

本设计基于 OPC 通信技术，实现了 MATLAB/Simulink 和 WinCC＋PLC 控制系统间的实时数据传输，并且直接调用 OPC Read/Write 功能，在 Simulink 中搭建控制算法实现了液位闭环控制。改变了以往 MATLAB 的纯数字仿真研究方法，也摈弃了用 MATLAB M 语言编写数据链接程序的做法。为控制研究人员通过 Simulink 实时观察和控制现场数据，进行控制算法的研究和控制结果的分析提供了一个平台。

在这个平台上，可以将 PID 算法改进为先进控制算法，比如 PID 自整定算法及模糊控制 PID 算法等。待算法成熟后，再将算法思想在 PLC 中用其编程语言编程实现，达到控制算法研究结果直接指导实际控制系统的目的。

习题

1. 简述计算机控制系统的设计原则。
2. 简述计算机控制系统的设计步骤。
3. 简述在做计算机控制系统的总体设计方案时,需要考虑哪些问题?
4. 简述计算机控制系统的硬件设计需要考虑哪些方面?
5. 简述计算机控制系统的软件设计需要考虑哪些方面?



课程实验

通过本课程的实验教学,使学生了解和掌握计算机控制的硬件技术和软件编程方法,对所学的理论知识有更深入的理解和认识,具有设计计算机控制系统的基本能力。

本课程设计的实验每次实验时间大约两个小时,其中的每一个实验学生都可以自己扩展,按照实验目的和要求自己设计,鼓励在实验基本要求的基础上创新,丰富实验内容。实验报告要求给出各个实验项目详细的设计内容,并叙述自己的实验体会。

11.1 模拟量过程通道和数据采集处理

11.1.1 实验目的

(1) 掌握用数据采集卡来采集模拟信号的硬件系统的接线方法。

(2) 掌握用数据采集卡来输出模拟量信号的硬件系统的接线方法。

(3) 深入学习 A/D 转换器原理及接口方法及 D/A 转换器的原理及接口方法。

(4) 掌握用 Kingview 组态王软件来编写数据采集卡的模拟量输入以及输出程序的方法。

11.1.2 实验设备

1. 硬件环境

工业控制计算机一台,PCI-1710HG 数据采集板卡、ADAM-3968 接线端子板以及 PCL-10168,0～5V 的可变电压源。

2. 软件平台

操作系统:Windows 2000 以上。

软件工具:Kingview 组态王软件及 PCI-1710 板卡的驱动程序。

11.1.3 实验内容与步骤

(1) 首先是正确地安装数据采集板卡。将 PCI-1710 板卡安装在工控机的扩展槽里,并安装板卡的驱动程序和研华板卡的检测程序。

(2) 模拟量输入通道数据的采集。即改变输入端的模拟电压值,利用 Kingview 组态王软件编程以连续方式采集并显示当前电压值,同时绘制电压变化的实时曲线。

(3) 模拟量输出通道数据的输出。即可以在程序中人为地产生一个变化的数值(比如

范围是(0～5)),利用外接电路指示灯的亮度变化以及示波器可以显示当前模拟量输出通道转换结果的电压值,并同时在程序中绘制给定数据的变化曲线。

(4) 扩展模拟量输出通道实验。编写程序自动生成连续变化的数值,通过输出通道的D/A转换产生一个脉冲三角波形,并在外接线路的指示灯中观察其亮度是否随之交替变化,同时用示波器观察其波形。

11.2 数字量过程通道和数据采集处理

11.2.1 实验目的

(1) 掌握用数据采集卡输入数字量信号的硬件系统的接线方法。
(2) 掌握用数据采集卡输出数字量信号的硬件系统的接线方法。
(3) 深入学习数字量输入输出通道的组成原理及接口方法。
(4) 掌握用Kingview组态王软件来编写数据采集卡的数字量输入以及输出程序的方法。

11.2.2 实验设备

1. 硬件环境

工业控制计算机一台,PCI-1710HG数据采集板卡、ADAM-3968接线端子板以及PCL-10168,电气开关、继电器以及指示灯等。

2. 软件平台

操作系统:Windows 2000以上。
软件工具:Kingview组态王软件及PCI-1710板卡的驱动程序。

11.2.3 实验内容与步骤

(1) 首先是正确的安装数据采集板卡。将PCI-1710板卡安装在工控机的扩展槽里,并安装板卡的驱动程序和研华板卡的检测程序。

(2) 数字量输入通道数据的采集。即利用外部线路中的电气开关产生开关数字信号,在组态王程序中采集并通过程序使画面中信号指示灯的颜色随之改变。

(3) 数字量输出通道数据的输出。即通过在程序中人为地产生一个数字信号,利用外接继电器电路控制指示灯的亮灭来验证数字量的输出,并同时在画面中实现信号指示灯颜色的变化。

11.3 地址译码电路设计

11.3.1 实验目的

(1) 熟悉Proteus软件的开发环境。
(2) 学习3-8译码器在接口电路中的应用。

(3) 掌握I/O接口地址译码电路的一般设计方法。

11.3.2 实验设备

1. 硬件环境

微型计算机一台,Pentium 4以上各类微机。

2. 软件平台

操作系统:Windows 2000以上。
仿真软件工具:Proteus软件。

11.3.3 实验内容与步骤

(1) 使用ISA总线的AEN(DMA地址使用信号,高电平有效)信号、地址信号A0~A9、3-8译码器芯片74HCT138,以及其他的基本门电路,在Proteus中搭建一个地址译码电路,使其译码出的8个端口的可寻址的地址范围为3A0H~3A7H。

(2) 运行程序,观察设计的地址译码电路译出地址的正确性。

11.4 ADC0808模数转换器硬件设计实验

11.4.1 实验目的

(1) 熟悉Proteus软件的开发环境。
(2) 熟悉Keil软件的使用以及程序设计方法。
(3) 熟悉模数转换器的工作原理。

11.4.2 实验设备

1. 硬件环境

微型计算机一台,Pentium 4以上各类微机。

2. 软件平台

操作系统:Windows 2000以上。
仿真软件工具:Proteus软件以及Keil软件。

11.4.3 实验内容与步骤

(1) 在Proteus中完成如图11.1所示的硬件系统搭建,仿真中主要用到的器件为AT89C51单片机、ADC0808以及7SEG-MPX4-CA数码显示管*4。

(2) 在Keil中完成软件程序的编写。

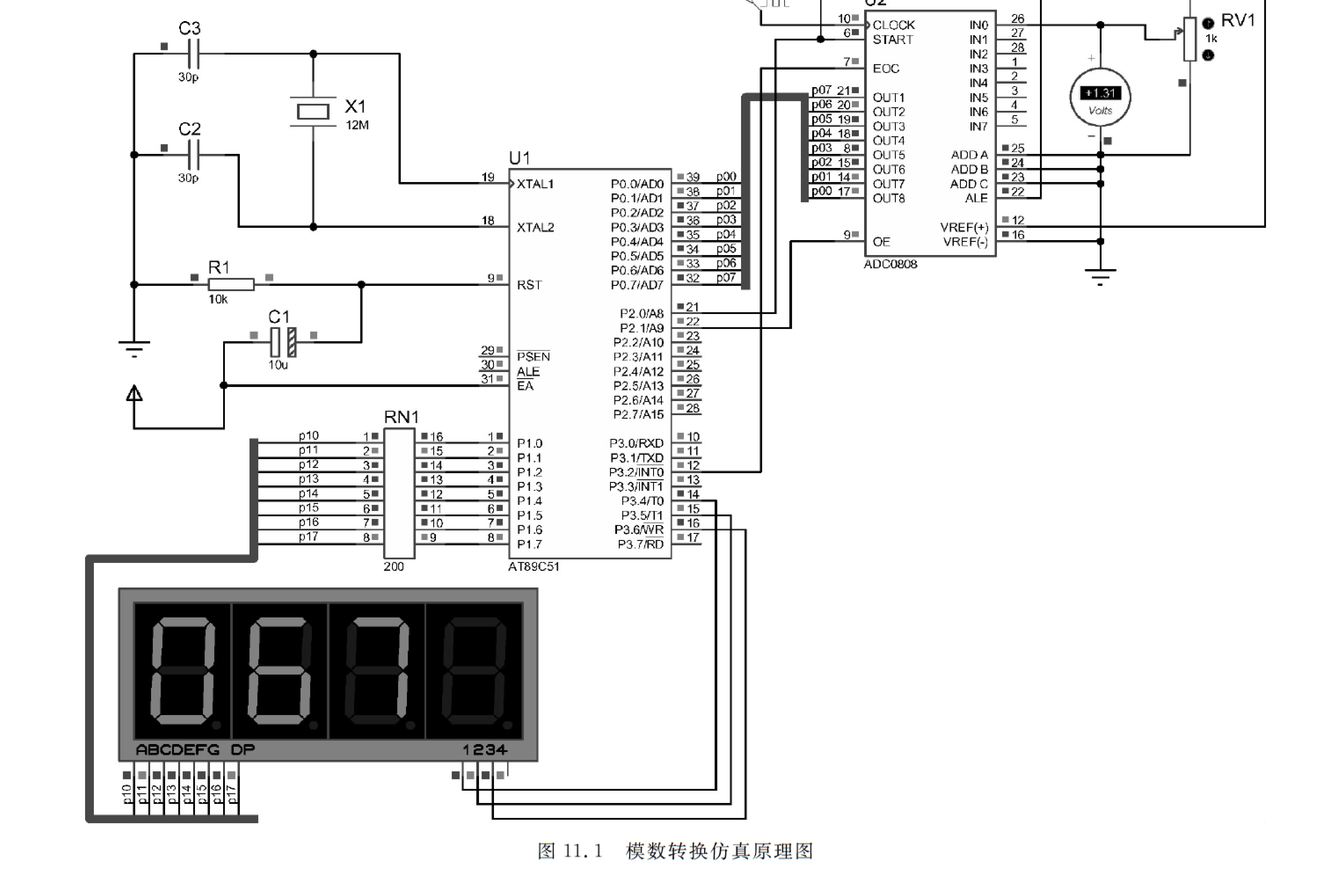


图 11.1 模数转换仿真原理图

(3) 进行软硬件的联合联调。

(4) 模数转换结果验证，比较电压表显示数据和数码管显示数据是否一致，并填写表 11.1。

表 11.1　模数转换结果比较表

	1	2	3	4	5	6	7	8
电压值								
二进制值								
十进制								

11.5　采样控制系统分析

11.5.1　实验目的

(1) 熟悉 MATLAB 中 Simulink 的实现方法。

(2) 考察连续时间系统的采样控制中，零阶保持器的作用与采样周期 T 对系统稳定性的影响。

11.5.2　实验设备

1. 硬件环境

微型计算机一台，Pentium 4 以上各类微机。

2. 软件平台

操作系统：Windows 2000 以上。

仿真软件工具：MATLAB 软件(6.1 以上版本)。

11.5.3　实验内容与步骤

(1) 开机执行 MATLAB 程序，用鼠标双击 MATLAB 图标进入其命令窗口 Command Windows，在 Command Windows 窗口中输入 Simulink，进入仿真界面，并新建 Model 文件。在 Model 界面中构造如图 11.2 所示的连续时间系统结构图。作时域仿真并确定系统时域性能指标。

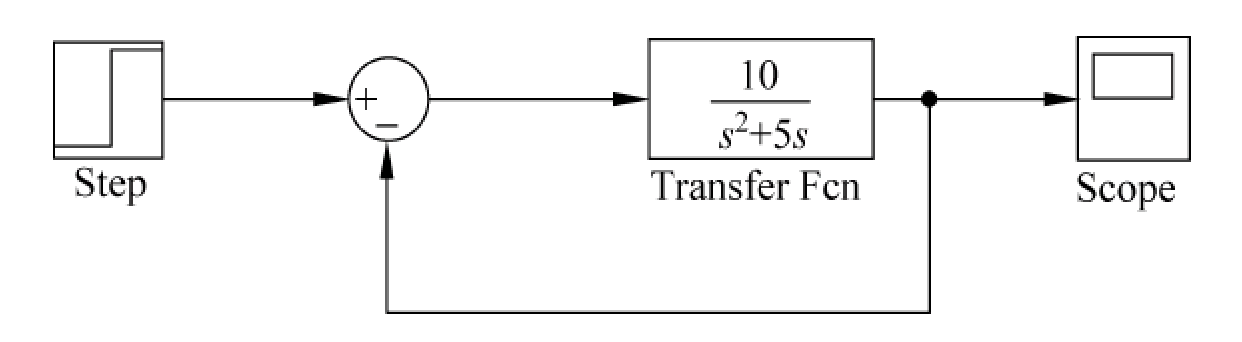


图 11.2　连续系统结构图

(2) 带零阶保持器的采样控制系统如图 11.3 所示。同样在 MATLAB 的 Simulink 下作时域仿真,并调整采样周期 T,观察对系统稳定性的影响。

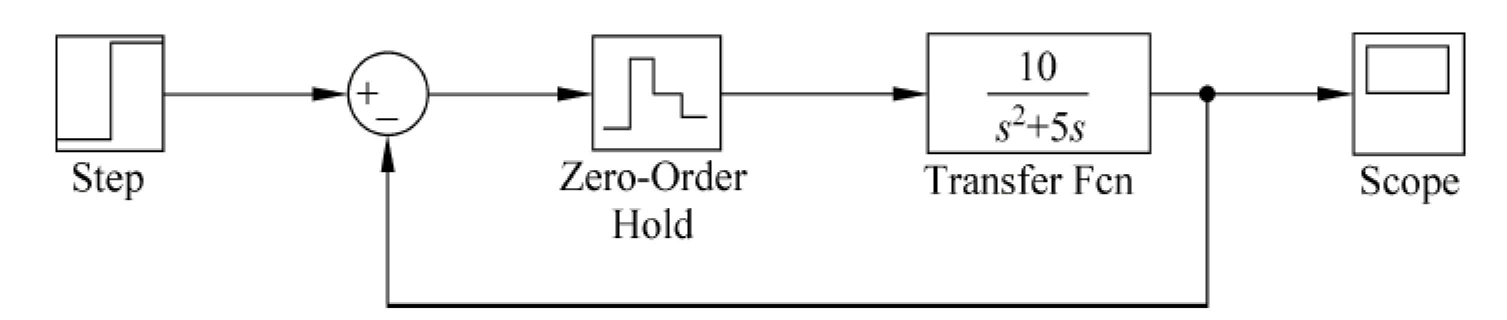


图 11.3 带零阶保持器的采样控制系统结构图

(3) 叙述零阶保持器的作用。

(4) 讨论采样周期 T 对系统特性的影响。

11.6 PID 调节器参数整定分析

11.6.1 实验目的

(1) 掌握 MATLAB 的 M 文件编写方法。

(2) 研究 PID 控制器的 K_p、T_i、T_d 三个参数对系统稳定性及过渡过程的影响。

11.6.2 实验设备

1. 硬件环境

微型计算机一台,Pentium 4 以上各类微机。

2. 软件平台

操作系统:Windows 2000 以上。

仿真软件工具:MATLAB 软件(6.1 以上版本)。

11.6.3 实验内容与步骤

(1) 开机执行 MATLAB 程序,用鼠标双击 MATLAB 图标进入 MATLAB 命令窗口 Command Windows,新建 M-file,然后分别输入以下设计好的程序。

① % 比例控制

```
G = tf(10,[1 6 11 6]); P = [0.01: 0.1: 1];        %传递函数,Kp 变化范围
for i = 1: length(P)
G_c = feedback(P(i) * G,1);                        %系统反馈
step(G_c),hold on                                  %单位阶跃输入响应
end
```

② % 比例积分控制

```
G = tf(10,[1 6 11 6])                              %系统传递函数
Kp = 1; Ti = [0.8: 0.2: 1.6];                      %Kp 一定,不同 Ti 的输入
for i = 1: length(Ti)
```

```
     Gc = tf(Kp * [1,1/Ti(i)],[1,0]);              % 分别计算不同 Ti 的系统参数
     G_c = feedback(G * Gc,1);
step(G_c),hold on                                   % 阶跃响应曲线响应
```

③ % 比例、积分、微分控制

```
G = tf(10,[1 6 11 6])                              % 系统传递函数
Kp = 1; Ti = 0.6; Td = [0.1: 0.2: 2];              % Kp、Ti 一定,不同 Td 的输入
for i = 1: length(Td)
     Gc = tf(Kp * [Ti * Td(i),Ti,1]/Ti,[1,0]);     % 分别计算不同 Td 的系统参数
     G_c = feedback(G * Gc,1); step(G_c),hold on    % 阶跃响应曲线
end
```

(2) 进行程序调试,并检查错误,然后运行。

(3) 观察系统对不同参数的响应曲线,分析其原因。

(4) 总结三个参数分别对系统性能的影响。

11.7 最少拍控制系统计算机仿真

11.7.1 实验目的

(1) 掌握最少拍有纹波、无纹波系统的设计方法。

(2) 学会最少拍控制系统的分析方法。

(3) 学习并掌握最少拍有纹波和无纹波控制器的设计以及 Simulink 的实现方法。

(4) 了解输入信号对最少拍控制系统的影响及其改进措施。

11.7.2 实验设备

1. 硬件环境

微型计算机一台,Pentium 4 以上各类微机。

2. 软件平台

操作系统:Windows 2000 以上。

仿真软件工具:MATLAB 软件(6.1 以上版本)。

11.7.3 实验内容与步骤

(1) 在如图 11.4 所示的计算机控制系统中,被控对象为 $G_c(s)=\dfrac{10}{s(s+1)}$,采样周期 $T=1\text{s}$,分别取输入信号为单位阶跃信号、单位速度信号和单位加速度信号,按有纹波系统和无纹波系统分别设计控制器 $D(z)$。

(2) 按照最少拍系统原理图,在 MATLAB 的 Simulink 下构造系统结构图的仿真模型,并将输入、输出和误差三条曲线放置在同一坐标系内,标明参数,观察输入输出波形,比较三种情况下的跟随特性,并打印结果。

(3) 观察系统输出波形在采样点以外的波形。

(4) 比较有纹波系统与无纹波系统的区别，并分析其原因。

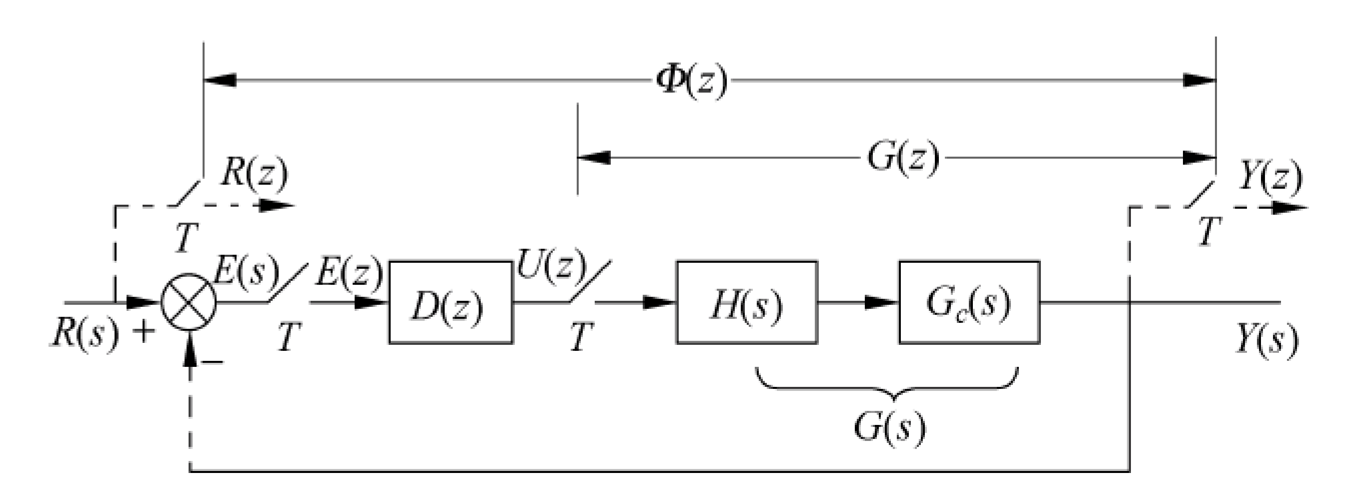


图 11.4 最少拍系统原理图

11.8 组态王软件的组态设计

11.8.1 实验目的

(1) 掌握并熟悉组态王软件的使用。

(2) 掌握研华 PCL-818L A/D 数据采集板卡的使用方法。

(3) 熟悉组态王软件与 PCL-818L 板卡的连接与通信。

(4) 熟悉 Pt100 温度传感器及变送器的使用。

11.8.2 实验设备

1. 硬件环境

工业控制计算机一台，PCL-818L 板卡及 PCLD-8115 接线端子板，Pt100 温度传感器及变送器。

2. 软件平台

操作系统：Windows 2000 以上。

软件工具：Kingview 组态王软件及 PCL-818L 板卡的驱动程序。

11.8.3 实验内容与步骤

(1) 首先是正确安装 A/D 数据采集板卡。将 PCL-818L 板卡进行硬件跳线配置，然后在计算机断电的情况下将 PLC-818L 板卡安装在研华工控机的扩展槽里，并安装研华 PLC-818L 板卡的驱动程序和研华板卡的检测程序。

(2) 进行 Pt100 温度传感器及变送器的安装，使其信号能正确地送入到 PLC-818L 板卡中。

(3) 启动组态王软件，进行监控画面设计，主要是构造数据库、建立动画连接以及监控界面的设计。

(4) 运行和调试实现温度采集系统的监控。

参 考 文 献

[1] 薛定宇.控制系统计算机辅助设计——MATLAB 语言与应用.北京：清华大学出版社，2006.
[2] 涂植英，陈今润.自动控制原理.重庆：重庆大学出版社，2005.
[3] 刘金琨.先进 PID 控制 MATLAB 仿真(第二版).北京：电子工业出版社，2005.
[4] 党宏社.控制系统仿真.西安：西安电子科技大学出版社，2008.
[5] 瞿亮.基于 MATLAB 的控制系统计算机仿真.北京：清华大学出版社，，2006.
[6] 李华，范多旺.计算机控制系统.北京：机械工业出版社，2007.
[7] 李江全.计算机控制技术与实训.北京：机械工业出版社，2010.
[8] 夏建全，等.工业计算机控制技术——原理与应用.北京：清华大学出版社，2006.
[9] 何克忠，李伟.计算机控制系统.北京：清华大学出版社，2007.
[10] 魏克新.自动控制综合应用技术——嵌入式微控制器、PLC、变频器、触摸屏、工控机、组态软件的综合应用.北京：机械工业出版社，2007.
[11] 亚控科技发展有限公司.组态王 6.53 使用手册.
[12] 林敏.计算机控制技术及工程应用.北京：国防工业出版社，2010.
[13] 刘恩博，等.组态软件数据采集与串口通信测控应用实战.北京：人民邮电出版社，2010.
[14] 高金源，夏洁.计算机控制系统.北京：清华大学出版社，2007.
[15] 于海生.微型计算机控制技术.北京：清华大学出版社，2009.
[16] 许勇.计算机控制技术.北京：机械工业出版社，2008.
[17] 邓健平.数控机床控制技术基础.北京：人民邮电出版社，2010.
[18] 于海生.计算机控制技术.北京：机械工业出版社，2011.
[19] 钱平.伺服系统.北京：机械工业出版社，2011.
[20] 敖荣庆，袁坤.伺服系统.北京：航空工业出版社，2006.
[21] 王建华，黄河清.计算机控制技术.北京：高等教育出版社，2003.
[22] 刘士荣.计算机控制系统.北京：机械工业出版社，2008.
[23] 范立南，李雪飞.计算机控制技术.北京：机械工业出版社，2009.
[24] 王平，肖琼.计算机控制系统.北京：高等教育出版社，2004.
[25] 高有堂，翟天嵩.电子设计与实战指导.北京：电子工业出版社，2007.
[26] 殷庆纵，李洪群.可编程控制器原理与实践.北京：清华大学出版社，2010.
[27] 朱清慧，张凤蕊，翟天嵩.Proteus 教程——电子线路设计、制版与仿真(第 2 版).北京：清华大学出版社，2011.
[28] 姜学军，刘新国.计算机控制技术.北京：清华大学出版社，2009.